Accepted Manuscript

An adaptive moving mesh method for thin film flow equations with surface tension

Abdulghani Alharbi, Shailesh Naire

PII: S0377-0427(17)30040-7
DOI: http://dx.doi.org/10.1016/j.cam.2017.01.019
Reference: CAM 10984

To appear in: *Journal of Computational and Applied Mathematics*

Received date: 12 April 2016
Revised date: 1 December 2016

Please cite this article as: A. Alharbi, S. Naire, An adaptive moving mesh method for thin film flow equations with surface tension, *Journal of Computational and Applied Mathematics* (2017), http://dx.doi.org/10.1016/j.cam.2017.01.019

1. We have successfully applied a r-adaptive moving mesh method based on MMPDEs and mesh density functions to a coupled system of higher order parabolic PDEs. To our knowledge this is the first attempt to implement r-adaptive schemes to such PDEs.

2. Numerical experiments show the adaptive moving mesh method to accurately resolve the multiple one-dimensional structures observed in the test problems. Moreover, it also reduces the computational effort in comparison to the numerical solution using the finite difference scheme on a fixed uniform mesh.

3. A significant result is related to adaptation of the curvature mesh density function to accurately resolve the solution at multiple locations using piecewise constant weight parameters.

4. We have also adapted the curvature mesh density function to include multiple solution components. This enabled us to accurately resolve the complicated multiple structures in the solution components compared to numerical solutions using a uniform mesh.

An adaptive moving mesh method for thin film flow equations with surface tension

Abdulghani Alharbi*, Shailesh Naire**

School of Computing and Mathematics, Keele University, Keele ST5 5BG, United Kingdom

Abstract

We present an adaptive moving mesh method for the numerical solution of thin liquid film spreading flows with surface tension. We follow the r-adaptive moving mesh technique which utilises a mesh density function and moving mesh partial differential equations (MMPDEs) to adapt and move the mesh coupled to the PDE(s) describing the thin film flow problem. Numerical experiments are performed on two one dimensional thin film flow equations to test the accuracy and efficiency of the method. This technique accurately resolves the multiple one-dimensional structures observed in these test problems. Moreover, it reduces the computational effort in comparison to the numerical solution using the finite difference scheme on a fixed uniform mesh.

Keywords: Thin film flows; surface tension; adaptive moving mesh; r-adaptive method; moving mesh PDEs (MMPDEs); mesh density function

1. Introduction

Thin liquid film flows driven by external forces, such as gravity or surface tension gradients (e.g., due to surfactant concentration gradients), are im-

*Present address: Mathematics Department, Taibah University, Medina, Kingdom of Saudi Arabia, Universities Road, PO Box: 344

**Corresponding author

Email address: s.naire@keele.ac.uk (Shailesh Naire)

portant in a wide range of industrial [1, 2], biomedical [3, 4], geophysical and environmental applications [5]. They display interesting dynamics, such as wave propagation and steepening, finite time singularities leading to film rupture and spatial "fingering" instabilities (see [6, 7] for an excellent review on these).

A particular class of thin film flow problems that is of particular interest both in the physical and mathematical context are those that include moving contact lines [8] and surface tension effects [2]. Typically in these problems, surface tension is only important in regions of very short length scales, particularly, near the contact lines, where the film's free surface exhibits internal layers with large spatial variation in the film's free surface curvature and away from these internal layers surface tension is relatively unimportant and the curvature is almost negligible. In such flows, the evolution of these internal layers has been associated with the onset of a transverse spatial fingering instability [9–15], hence their accurate resolution is important to understand the mechanisms behind this instability. For the numerical solution of these problems, one can then use a locally refined mesh in the regions of large spatial variation and a coarser mesh elsewhere. In contrast, a uniform mesh solution would use an unacceptably large number of mesh points especially due to the relatively large spatial scale and long time scale typical in spreading flows. The main motivation for this paper is to develop a numerical solution technique for thin film spreading flows which employs a moving and adaptive mesh that would accurately resolve such internal layers in a computationally efficient way compared to a fixed and uniform mesh.

A long wavelength or lubrication approximation is commonly employed to derive the thin film flow equations [6, 7]. When surface tension is included, this reduces the governing fluid flow equations and boundary conditions to a fourth order nonlinear parabolic PDE representing the evolution of the film's free surface [16]. In most problems, this may be coupled to a parabolic PDE (usually of second order), for example, representing the surfactant concentration. In the context of thin film spreading flows, there have been numerous numerical experiments using the finite difference method on a fixed uniform or nonuniform mesh in both one and two dimensions ([10–15, 17–24], to name a few). Other methods that have also been applied include the finite element method [25, 26] and spectral methods [27]. In comparison, very few numerical studies have considered an adaptive mesh. The works of Warner *et al.* [12, 19], Edmonstone *et al.* [13, 20, 21] and Mavromoustaki *et al.*

[24] use general purpose publicly available solvers for parabolic PDEs which have built-in adaptive mesh capabilities (PDECOL [28] based on collocation methods; TOMS731 [29–32] based on a monitor function to adapt mesh and a Lagrangian method for moving the mesh points). Sun *et al.* [33] use a h-adaptive finite element mesh refinement method based on an optimal interpolation error estimate for a two dimensional thin film equation of gravity driven flow down an inclined plane. Li *et al.* [34] have also developed a h-adaptive finite difference method for this equation using a fully discrete and nonlinear multigrid scheme and adaptive mesh refinement method. The above adaptive mesh schemes were able to capture and resolve the moving contact line and the associated fingering instability accurately and computationally efficiently compared to a fixed uniform grid scheme. Lee *et al.* [35, 36] solved a thin film flow problem over a plane containing single and grouped topographic features using a full approximation storage multigrid algorithm and employing automatic mesh adaptivity.

In this paper, we follow the r-adaptive moving mesh technique [37, 38] which utilises a mesh density function and moving mesh partial differential equations (MMPDEs) to adapt and move the mesh coupled to the PDE(s) describing the thin film flow problem. We believe that r-adaptive schemes are better suited for thin film flow equations compared to the h (or hp)-adaptive schemes mentioned above. The performance of h-adaptive schemes depends on the availability of a good *a posteriori* indicator of the solution error. In contrast, r-adaptive schemes refine the mesh based on a mesh density function that directly tracks a particular solution characteristic (for example, the curvature of the film's free surface) and hence could resolve the internal layers more accurately. The MMPDEs usually take the general form of a nonlinear diffusion equation [37, 38] which is coupled to the thin film equations. This can be easily implemented within the existing finite difference framework well suited for these problems. h-adaptive schemes, on the other hand, are usually implemented using the finite element method. Although r-adaptive methods are a recent development and have not been used as frequently as h or hp- refinements, they have been successfully applied in various applications, such as computational fluid mechanics [39], convective heat transfer [40] and meteorological [41, 42] problems. The r-refinement techniques also form the basis of general purpose publicly available solvers for one dimensional parabolic PDE systems, for example, MOVCOL [43] and TOMS731 [32]. However, they have been only tested for second order parabolic PDEs

(for example, the Burger's equation) and to our knowledge this is the first attempt to implement r-adaptive schemes and MMPDEs to higher order parabolic PDEs.

As a test problem, we consider the spreading of a surfactant laden liquid drop or sheet down an inclined substrate prewetted with a precursor liquid film due to gravity and surface tension gradients (due to gradients in surfactant concentration). Previous one dimensional numerical simulations and mathematical analysis have revealed multiple propagating wave like structures joined together by internal layers where surface tension is important [13, 15, 20–24]. Numerical simulations of the two dimensional flow show a transverse spatial fingering instability that develops near the moving effective contact line (where the drop connects onto the precursor film ahead of it) [13, 20, 21]. In this paper we focus on implementing the r-adaptive moving mesh method for the one dimensional problem; extending this to two dimensions is currently being undertaken and will be reported elsewhere.

The rest of the paper is organized as follows. In section §2, we provide a brief overview of moving mesh partial differential equations (MMPDEs) and mesh density functions. In §3, we briefly describe the governing equations and boundary conditions for one dimensional gravity and surface tension gradient driven thin liquid film flow using a lubrication theory model. In §4, the spatial discretisation of the governing equations and the MMPDEs using the finite difference method is presented. In §5, we present the numerical results. Conclusions are given in §6.

2. Moving Mesh Partial Differential Equations (MMPDEs) and mesh density functions

The underlying principle behind r-adaptive moving mesh methods considered here is that the mesh moves continuously in time while adapting to the evolving structures in the solution. Using the equidistribution principal (in which a continuous function defined over an interval is evenly distributed between the subintervals determined by the mesh points), a number of moving mesh equations (MMPDEs) can be developed for time dependent problems. These are continuous forms of mesh movement strategies formulated in terms of coordinate transformations [37, 38]. The mesh adapts itself based on a mesh density function which is related to a specific solution characteristic.

In one dimension, the MMPDEs can be represented as a coordinate transformation:

$$x = x(\xi, t) \colon \xi \in \Omega_c \equiv [0,1] \rightarrow x \in \Omega_p \equiv [a,b],\ t > 0,$$

where x and ξ are the spatial variables, t is time, Ω_c and Ω_p are referred to as the computational and physical domains, respectively, and a fixed uniform mesh is used to represent any discretisation of Ω_c. We list four of the commonly used one dimensional MMPDEs, the so-called MMPDEs 4, 5 and 6 and modified MMPDE5. For details of their derivations and other theoretical considerations, the interested reader is referred to the book, *Adaptive Moving Mesh Methods*, by Huang & Russell [37] and the review paper by Budd *et al.* [38], who have made seminal contributions in this area over the past twenty years.

$$\text{MMPDE4}: \quad (\hat{\rho} x_{t\xi})_\xi = -\frac{1}{\tau}(\hat{\rho} x_\xi)_\xi, \tag{1}$$

$$\text{MMPDE5}: \quad x_t = \frac{1}{\tau}(\hat{\rho} x_\xi)_\xi, \tag{2}$$

$$\text{MMPDE6}: \quad x_{t,\xi\xi} = -\frac{1}{\tau}(\hat{\rho} x_\xi)_\xi, \tag{3}$$

$$\text{modified MMPDE5}: \quad x_t = \frac{1}{\hat{\rho}\tau}(\hat{\rho} x_\xi)_\xi. \tag{4}$$

Here, $\hat{\rho}(x,t)$ is a mesh density function (defined below) and $\tau > 0$ is a user-specified parameter. τ adjusts the response time of mesh movement to changes in the monitor function $\hat{\rho}(x,t)$ [37]. The smaller τ, the more quickly the mesh responds to changes in $\hat{\rho}(x,t)$. Likewise, the mesh moves slowly when a large value of τ is used. MMPDE5 and modified MMPDE5 given in Eqs. (2),(4) are generally quite stiff and a regularised form is used in practice,

$$\text{regularised MMPDE5}: \quad x_t - \gamma_1 x_{t,\xi\xi} = \frac{1}{\tau}(\hat{\rho} x_\xi)_\xi, \tag{5}$$

$$\text{regularised modified MMPDE5}: \quad x_t - \gamma_1 x_{t,\xi\xi} = \frac{1}{\hat{\rho}\tau}(\hat{\rho} x_\xi)_\xi. \tag{6}$$

Here, the parameter $\gamma_1 > 0$ is related to the mesh density function $\hat{\rho}$ (see [38] and references therein). The boundary conditions for the above second

order PDEs are

$$x(0,t) = 0, \quad x(1,t) = b. \tag{7}$$

The initial condition is

$$x(\xi, 0) = (b-a)\xi + a, \tag{8}$$

which represents a uniform initial mesh on the physical domain $\Omega_p \equiv [a, b]$. We note here that a nonuniform initial mesh can also be used which is obtained by solving in pseudo time the chosen MMPDE (with the uniform mesh as the initial condition) with the solution fixed (hence, $\hat{\rho}$ is also fixed) at its initial condition.

The choice of the mesh density function $\hat{\rho}$ is essential for the success of adaptive moving mesh methods. They can be chosen based on error estimates (for example, polynomial interpolation or truncation error) or on the solution characteristics of the underlying PDE (for example, arc length or curvature). In the latter case, the mesh density function can be defined by the solution $u(x,t)$ (say) of the underlying PDE and possibly its derivatives. The choices of $\hat{\rho}$ in one dimension that are commonly used are:

$$\text{arc length}: \ \hat{\rho}(x,t) = \sqrt{1 + \alpha u_x^2}, \tag{9}$$

$$\text{curvature}: \quad \hat{\rho}(x,t) = \left(\alpha + \beta u_{xx}^2\right)^{\frac{1}{n}}, \ \text{where} \ \ n = 2 \text{ or } 4, \tag{10}$$

$$\text{optimal}: \ \hat{\rho}(x,t) = \left(1 + \frac{1}{\alpha} u_{xx}^2\right)^{1/3}, \ \ \alpha = \left[\frac{1}{a-b}\int_a^b u_{xx}^2 dx\right]. \tag{11}$$

Here, α and β are adaptivity parameters (or weight parameters) of the mesh density function [37, 44, 45]. These are usually taken to be constant but could be dependent on the spatial variable x if there are multiple regions over which the solution characteristics vary rapidly (see example in §5). The above monitor functions can be extended to include multiple solution components (see example in §5). In addition, it is common practice in the context of moving mesh methods to smooth the monitor function in order to obtain a smoother mesh and also to make the MMPDEs easier to integrate. This is discussed in §4.

3. Thin film equations for gravity and surfactant-related spreading

We consider the one-dimensional thin liquid film flow of a surfactant-laden drop or sheet spreading down an inclined and pre-wetted substrate due to gravity and surfactant-related effects (see Fig. 1). The bulk flow is governed

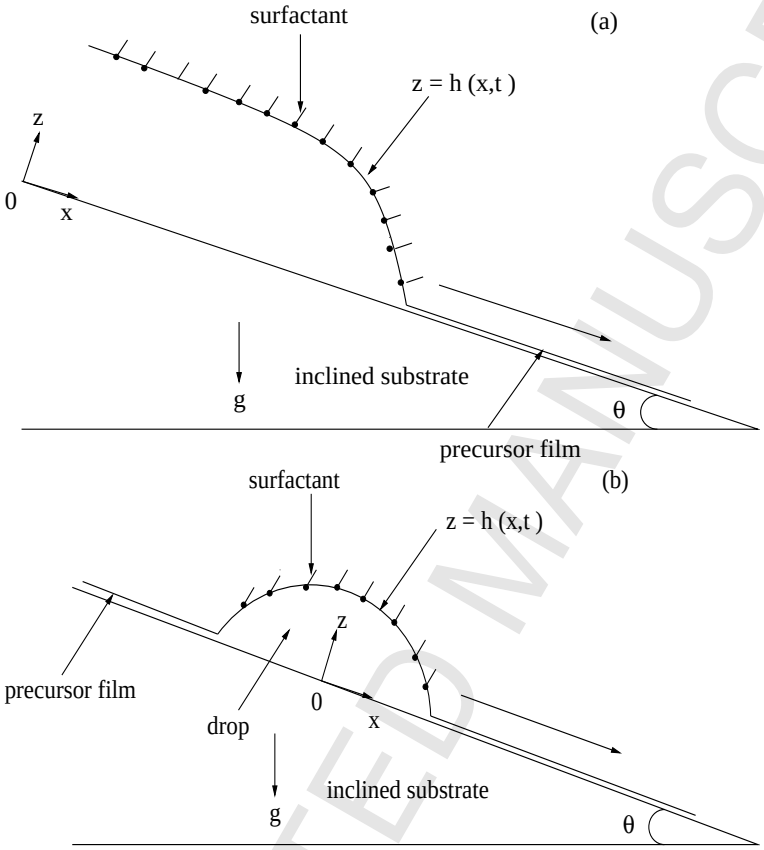

Figure 1: Schmatic of a surfactant-laden drop or sheet spreading down an inclined and pre-wetted substrate using (a) constant flux and (b) constant volume boundary conditions.

by the Navier-Stokes equations. The surfactant is considered insoluble (i.e., it is present only on the free surface) and its transport on the free surface is modelled using an advection-diffusion equation. We also assume that the substrate is pre-wetted with a thin precursor liquid film. Lubrication (or long wavelength) theory can be applied to reduce the governing equations and boundary conditions at the free surface to give the evolution equations for the film's free surface and the surfactant concentration. The interested

reader can refer to [13, 20, 46, 47] for their derivation. These can be written in non-dimensional form as:

$$h_t + \left[Ca\frac{h^3}{3}h_{xxx} - G\cos(\theta)\frac{h^3}{3}h_x - \frac{h^2}{2}\Gamma_x + G\sin(\theta)\ \frac{h^3}{3}\right]_x = 0, \tag{12}$$

$$\Gamma_t + \left[Ca\frac{h^2}{2}\Gamma h_{xxx} - G\cos(\theta)\frac{h^2}{2}\Gamma h_x - h\Gamma\Gamma_x + G\sin(\theta)\ \frac{h^2}{2}\Gamma\right]_x = \delta\Gamma_{xx}. \tag{13}$$

Here, $h = h(x,t)$ and $\Gamma = \Gamma(x,t)$ are the film height and surfactant concentration, respectively, x is the spatial variable and t is time. The dimensionless parameter Ca, is the inverse Capillary number (relates surface tension and viscous forces), G is a Bond number (relates gravity and viscous forces), δ, is the inverse of the Péclet number (compares the magnitude of changes to surfactant concentration due to diffusion and advection by surface flow) and θ is the substrate inclination angle. Eqs. (12),(13) are two coupled nonlinear parabolic PDEs of fourth and second order in space, respectively. The fourth order term in Eq. (12) is related to the curvature of the film's free surface and is due to surface tension. Typically $Ca \ll 1$, so there exist internal layers where curvature is important.

We prescribe two sets of boundary conditions (BCs) for the above PDEs, namely constant flux and constant volume. Constant flux BCs correspond to constant supply of fluid and surfactant at one end of the domain (see Fig. 1(a)) while constant volume prescribes no flux BCs at both ends of the domain (see Fig. 1(b)). We also assume that the plane is pre-wetted with a precursor film of thickness $b \ll 1$ (represents ratio of precursor film thickness to initial drop or sheet thickness) and it is surfactant free. These boundary conditions are prescribed as follows: The constant flux boundary conditions are given by:

$$h(0,t) = \Gamma(0,t) = 1, h(L,t) = b, \Gamma(L,t) = 0, h_{xxx}(0,t) = h_{xxx}(L,t) = 0, \tag{14}$$

where L is the length of the computational domain. These BCs represent a fixed fluid and surfactant flux source far upstream and a flat precursor film and zero surfactant concentration far downstream. The constant volume boundary conditions are given by

$$h(\pm L,t) = b, h_{xxx}(\pm L,t) = 0, \Gamma(\pm L,t) = 0, \tag{15}$$

where $\pm L$ are the boundaries of the physical domain.

The initial conditions for h and Γ corresponding to constant flux boundary conditions are given by (following [13, 20, 46, 47]):

$$h(x,0) = (1-x^2-b)H(1-x)+b, \;\; \Gamma(x,0) = H(1-x), \tag{16}$$

where $H(x)$ is the Heaviside function. The initial condition for h represents a drop connecting onto a thin precursor film and for the surfactant concentration represents a large gradient at the location where the drop connects onto the precursor film. The initial conditions for h and Γ corresponding to constant flux boundary conditions are given by (following [13, 20, 46, 47]):

$$h(x,0) = (1-x^2)[H(1-x)-H(-1-x)]+b[H(x-1)+H(-1-x)], \tag{17}$$
$$\Gamma(x,0) = H(1-x)-H(-x-1), \tag{18}$$

where $x = \pm 1$, is the initial location where the parabolic-shaped drop connects to the precursor film both upstream and downstream. The initial condition for Γ assumes a uniform surfactant concentration between $x = \pm 1$ and zero surfactant concentration ahead and behind it and is characterised by a steep gradient at $x = \pm 1$.

4. Finite difference semi-discretisation scheme on a moving adaptive mesh

The adaptive moving mesh method uses a coordinate transformation from the computational domain with coordinate ξ, to the physical domain with coordinate x: $x = x(\xi,t)\colon \Omega_c \equiv [0,1] \rightarrow \Omega_p \equiv [0,L]$, $t > 0$. Then the solution can be written as: $(h,\Gamma)(x,t) = (h,\Gamma)(x(\xi,t),t)$. A moving mesh on the physical domain associated with the solutions h and Γ is described as: $\mathcal{J}^p_{h,\Gamma}(t)\colon x_j(\xi) = x(\xi_j,t), j = 1,\ldots,N+1$, where the boundary nodes are given by: $x_1 = 0$, $x_{N+1} = L$. A uniform mesh on the computational domain is described as: $\mathcal{J}^c_{h,\Gamma}\colon \xi_j = (j-1)L/N, j = 1,\cdots,N+1$. Using the chain rule, Eqs. (12),(13) can be written in the computational coordinate ξ as:

$$h_t - \frac{h_\xi}{x_\xi}x_t = -\frac{Q_\xi}{x_\xi}, \quad \Gamma_t - \frac{\Gamma_\xi}{x_\xi}x_t = -\frac{S_\xi}{x_\xi}, \tag{19}$$

$$Q = Ca\left(\frac{h^3}{3}\right)\frac{1}{x_\xi}\left(\frac{\left(\frac{h_\xi}{x_\xi}\right)_\xi}{x_\xi}\right)_\xi - \beta_1\left(\frac{h^3}{3}\right)\frac{h_\xi}{x_\xi} - \left(\frac{h^2}{2}\right)\frac{\Gamma_\xi}{x_\xi} + \alpha_1\frac{h^3}{3}, \quad (20)$$

$$S = Ca\left(\frac{h^2\Gamma}{2}\right)\frac{1}{x_\xi}\left(\frac{\left(\frac{h_\xi}{x_\xi}\right)_\xi}{x_\xi}\right)_\xi - \beta_1\left(\frac{h^2\Gamma}{2}\right)\frac{h_\xi}{x_\xi} - (h\Gamma+\delta)\frac{\Gamma_\xi}{x_\xi} + \alpha_1\frac{h^2\Gamma}{2}. \quad (21)$$

In the above, $\alpha_1 = G\sin(\theta)$ and $\beta_1 = G\cos(\theta)$.

A *conservative* finite difference semi-discretisation scheme for the spatial derivatives in Eqs. (19)-(21) on the uniform mesh $\mathcal{J}^c_{h,\Gamma}$ using centred finite differences can be written as, keeping the time derivative continuous,

$$h_{t,j} - \frac{(\Delta h_j - \Delta h_{j-1})}{(\Delta x_j - \Delta x_{j-1})}x_{t,j} = -2\frac{\Delta Q_{j-1}}{\Delta x_j - \Delta x_{j-1}}, \quad j = 2,\ldots,N, \quad (22)$$

$$\Gamma_{t,j} - \frac{(\Delta \Gamma_j - \Delta \Gamma_{j-1})}{(\Delta x_j - \Delta x_{j-1})}x_{t,j} = -2\frac{\Delta S_{j-1}}{\Delta x_j - \Delta x_{j-1}}, \quad j = 2,\ldots,N, \quad (23)$$

where $\Delta(x_j, h_j, \Gamma_j, Q_j, S_j) = (x_{j+1}-x_j, h_{j+1}-h_j, Q_{j+1}-Q_j, S_{j+1}-S_j)$.

$$Q_j = Ca\left(\frac{h^3}{3}\right)_{j+1/2}\frac{1}{x_{\xi,j}}\left(\frac{\left(\frac{h_\xi}{x_\xi}\right)_\xi}{x_\xi}\right)_{\xi,j} - \beta_1\left(\frac{h^3}{3}\right)_{j+1/2}\left(\frac{h_\xi}{x_\xi}\right)_{,j} - \left(\frac{h^2}{2}\right)_{j+1/2}\left(\frac{\Gamma_\xi}{x_\xi}\right)_{,j} + \alpha_1\left(\frac{h^3}{3}\right)_{j+1/2}, \quad j = 1,\ldots,N, \quad (24)$$

$$S_j = Ca\left(\frac{h^2\Gamma}{2}\right)_{j+1/2}\frac{1}{x_{\xi,j}}\left(\frac{\left(\frac{h_\xi}{x_\xi}\right)_\xi}{x_\xi}\right)_{\xi,j} - \beta_1\left(\frac{h^2\Gamma}{2}\right)_{j+1/2}\left(\frac{h_\xi}{x_\xi}\right)_{,j} - \left[(h\Gamma)_{j+1/2}+\delta\right]\left(\frac{\Gamma_\xi}{x_\xi}\right)_{,j} + \alpha_1\left(\frac{h^2\Gamma}{2}\right)_{j+1/2}, \quad j = 1,\ldots,N. \quad (25)$$

The spatial derivatives appearing in Q_j and S_j are discretised as follows:

$$\frac{1}{x_{\xi,j}}\left(\frac{\left(\frac{h_\xi}{x_\xi}\right)_\xi}{x_\xi}\right)_{\xi,j} = \left(\frac{2}{\Delta x_j - \Delta x_{j-1}}\right)\left(\frac{1}{x_{\xi,j+1}}\left(\frac{h_\xi}{x_\xi}\right)_{\xi,j+1} - \frac{1}{x_{\xi,j}}\left(\frac{h_\xi}{x_\xi}\right)_{\xi,j}\right),$$

$$\frac{1}{x_{\xi,j}}\left(\frac{h_\xi}{x_\xi}\right)_{\xi,j} = \left(\frac{2}{\Delta x_j - \Delta x_{j-1}}\right)\left(\frac{\Delta h_j}{\Delta x_j} - \frac{\Delta h_{j-1}}{\Delta x_{j-1}}\right),$$

$$\left(\frac{h_\xi}{x_\xi}\right)_{,j} = \frac{\Delta h_j}{\Delta x_j}, \quad \left(\frac{\Gamma_\xi}{x_\xi}\right)_{,j} = \frac{\Delta \Gamma_j}{\Delta x_j}, \quad j = 1, \ldots, N. \tag{26}$$

In the above, we approximate terms denoted by $(a)_{j+\frac{1}{2}}$ using spatial averages: $(a)_{j+1/2} = (a_{j+1} + a_j)/2$ or $(a)_{j+1/2} = a(x_{j+1/2})$, where $x_{j+1/2} = (x_{j+1} + x_j)/2$. A similar approximation is used for terms denoted by $(a)_{j-\frac{1}{2}}$. It has been shown that there are particular choices for the approximation of $(h^3)_{j+\frac{1}{2}}$ that have special properties, for example, the so-called positivity-preserving scheme, meaning that if one starts from strictly positive data for h, the scheme will help preserving this property. The interested reader is referred to the works by Zhornitskaya & Bertozzi [48] for details. Evaluating Eqs. (22),(23) at $j = 2, N$ require fictitious points $(h, x)_0$ and $(h, x)_{N+2}$. $h_{0,N+2}$ are obtained by discretizing the boundary conditions $h_{xxx} = 0$ at $x = 0, L$ using centred finite differences. We assume that the adaptive mesh is uniform near $x = 0, L$ (i.e., $x_\xi = 1$), so $x_{0,N+2}$ are obtained by discretizing this using centred finite differences. The boundary conditions $h(0,t) = \Gamma(0,t) = 1$, $h(L,t) = b$ and $\Gamma(L,t) = 0$ are replaced by their ODE form: $h_{t,1} = 0$, $h_{t,N+1} = 0$, $\Gamma_{t,1} = 0$, $\Gamma_{t,N+1} = 0$.

Other semi-discretisation finite difference schemes have also been considered for the spatial derivatives in Eqs. (12),(13). For example, in Appendix A we write a *non-conservative* scheme. In all the numerical experiments we performed, the semi-discretisation scheme given in Eqs. (22)-(26) gave the best performance in terms of accuracy and convergence, and ensured conservation of fluid volume. We only report results based on the conservative scheme in §5.

The equidistributing coordinate transformation $x = x(\xi, t)$ is obtained by solving the MMPDEs given in Eqs. (1)-(6). A semi-discretisation scheme using centred finite differences to discretise the spatial derivatives in Eqs.

(1)-(6) is as follows, keeping the time derivative continuous:

$$\text{MMPDE4}: \ \tau[\hat{\rho}_{j+1/2}\Delta x_{t,j} - \hat{\rho}_{j-1/2}\Delta x_{t,j-1}] = -[\hat{\rho}_{j+1/2}\Delta x_j - \hat{\rho}_{j-1/2}\Delta x_{j-1}], \quad j = 2, \ldots, N. \tag{27}$$

$$\text{MMPDE5}: \ \tau x_{t,j} = \frac{1}{\Delta\xi^2}[\hat{\rho}_{j+1/2}\Delta x_j - \hat{\rho}_{j-1/2}\Delta x_{j-1}], \ j = 2, \ldots, N. \tag{28}$$

$$\text{MMPDE6}: \ \tau(\Delta x_{t,j} - \Delta x_{t,j-1}) = -[\hat{\rho}_{j+1/2}\Delta x_j - \hat{\rho}_{j-1/2}\Delta x_{j-1}], \quad j = 2, \ldots, N. \tag{29}$$

$$\text{modified MMPDE5}: \ \tau x_{t,j} = \frac{1}{\hat{\rho}_j\Delta\xi^2}[\hat{\rho}_{j+1/2}\Delta x_j - \hat{\rho}_{j-1/2}\Delta x_{j-1}], \quad j = 2, \ldots, N. \tag{30}$$

The regularised MMPDE5 and modified MMPDE5 are discretised as

$$\text{regularised MMPDE5}: \ \tau[x_{t,j} - \frac{\gamma_1}{2\Delta\xi^2}(\Delta x_{t,j} - \Delta x_{t,j-1})] = \frac{1}{\Delta\xi^2}[\hat{\rho}_{j+1/2}\Delta x_j - \hat{\rho}_{j-1/2}\Delta x_{j-1}], \ j = 2, \ldots, N. \tag{31}$$

$$\text{regularised modified MMPDE5}: \ \tau[x_{t,j} - \frac{\gamma_1}{2\Delta\xi^2}(\Delta x_{t,j} - \Delta x_{t,j-1})] = \frac{1}{\hat{\rho}_j\Delta\xi^2}[\hat{\rho}_{j+1/2}\Delta x_j - \hat{\rho}_{j-1/2}\Delta x_{j-1}], \quad j = 2, \ldots, N. \tag{32}$$

In the above, $\Delta x_{t,j} = x_{t,j+1} - x_{t,j}$, $\hat{\rho}_{j+1/2} = (\hat{\rho}_{j+1} + \hat{\rho}_j)/2$ and $\hat{\rho}_{j-1/2} = (\hat{\rho}_{j-1} + \hat{\rho}_j)/2$.

The mesh density function $\hat{\rho}(x,t)$ given in Eqs (9),(10) are discretised using

finite differences as:

$$\text{Arc length mesh density function}: \ \hat{\rho}(x_j,t) = \sqrt{\alpha + \beta h_{x,j}^2}, \tag{33}$$
$$h_{x,j} := \begin{cases} (\Delta h_j - \Delta h_{j-1})/(\Delta x_j - \Delta x_{j-1}), & j = 2,\ldots,N, \\ \Delta h_1/\Delta x_1, & j = 1, \\ \Delta h_N/\Delta x_N, & j = N+1. \end{cases}$$

$$\text{Curvature mesh density function}: \hat{\rho}(x_j,t) = (\alpha + \beta h_{xx,j}^2)^{\frac{1}{n}}, \ \ n = 2,4 \tag{34}$$
$$h_{xx,j} := \begin{cases} [2/(\Delta x_j + \Delta x_{j-1})]\left[\Delta h_j/\Delta x_j - \Delta h_{j-1}/\Delta x_{j-1}\right], & j = 2,...,N, \\ 2\left[\Delta x_1(\Delta h_2 + \Delta h_1) - (\Delta x_2 + \Delta x_1)(\Delta h_1)\right]/ \\ \left[(\Delta x_2 + \Delta x_1)\Delta x_2 \Delta x_1\right], & j = 1, \\ -2\left[\Delta x_N(\Delta h_N + \Delta h_{N-1}) - (\Delta x_N + \Delta x_{N-1})\Delta h_N\right]/ \\ \left[(\Delta x_N + \Delta x_{N-1})\Delta x_N \Delta x_{N-1}\right], & j = N+1. \end{cases}$$

To obtain a smoother mesh and also make the MMPDEs easier to integrate, it is common practice in the context of moving mesh methods to smooth the mesh density function. A simple but effective smoothing scheme suggested by Huang [37, 49] is based on weighted averaging,

$$\hat{\rho}_j := \sqrt{\frac{\sum_{k=j-p}^{j+p} \hat{\rho}_k^2 \left(\frac{\gamma}{1+\gamma}\right)^{|k-j|}}{\sum_{k=j-p}^{j+p} \left(\frac{\gamma}{1+\gamma}\right)^{|k-j|}}}, \quad j = 1,\ldots,N+1, \tag{35}$$

where p is a non-negative integer called the smoothing index and γ is a positive smoothing parameter. Several sweeps of the scheme may be applied at each integration step.

Eqs. (22),(23) and Eqs. (27)-(32), form a coupled system of $3(N+1)$ ordinary differential equations (ODEs) for the solution $h_1,\ldots,h_{N+1}$, $\Gamma_1,\ldots,\Gamma_{N+1}$ and the mesh $x_1,\ldots,x_{N+1}$, with initial conditions for h and Γ given by Eq. (16) or Eqs. (17), (18) and $x(\xi,0) = L\xi$. A non-uniform initial mesh was also used which was obtained by solving in pseudo-time the chosen MMPDE (with the uniform mesh as the initial condition) with h and Γ fixed (hence, the mesh density function $\hat{\rho}$ is also fixed) at its initial condition. This did not have any significant influence on the solution or the performance of the solver in comparison to the uniform initial mesh. We solve the resulting ODEs simul-

taneously using the stiff ODE solver DASSL [50] or DASPK [51]. These solvers use either a direct (DASSL) or iterative (DASPK - based on preconditioned Krylov subspace method) method to solve the linearised system of equations. DASPK also allows approximating the Jacobian using an Incomplete LU factorisation. This has a significant influence on the performance in comparison to DASSL. However, we need to choose a sufficiently large fill-in for the LU factorisation of the Jacobian, otherwise the convergence of the iterative solver is very slow. We use a staggered system for numbering the unknowns, $h_1, \Gamma_1, x_1, h_2, \Gamma_2, x_2, \ldots, h_{N+1}, \Gamma_{N+1}, x_{N+1}$, which provides a smaller bandwidth for the Jacobian matrix (note: the bandwidth=13 for the semi-discretisation in Eqs. (22),(23); it could be larger if smoothing of mesh density function is used. This is in comparison to, for example, the numbering $h_1, h_2, \ldots, h_{N+1}, \Gamma_1, \Gamma_2, \ldots, \Gamma_{N+1}, x_1, x_2, \ldots, x_{N+1}$, which although sparse has a much bigger bandwidth. This significantly improves the performance of the ODE solver.

5. Numerical results

In this section, we perform numerical experiments on the coupled thin film spreading flow problem for the free surface thickness h and surfactant concentration Γ given by Eqs. (12),(13). In all the results presented below, the system parameter values are: $Ca = 10^{-3}$ (indicating smaller surface tension relative to viscous forces), $\theta = 90^o$ (representing a vertical substrate), $G = 1$, (so, $\alpha_1 = 0$, $\beta_1 = 1$), $b = 10^{-2}$ (precursor film thickness is 1/100 times smaller than the initial drop or sheet thickness) and $\delta = 10^{-5}$ (indicating surfactant diffusion is very small). The mesh density function smoothing parameters are $p = 2$ and $\gamma = 2$.

We first consider only the gravity-driven flow problem (by setting $\Gamma = 0$ in Eqs. (12),(13)) which reduces the system to a single fourth order parabolic PDE for h. A travelling wave solution exists for this problem when constant flux boundary conditions (Eq. (14)) are applied [11, 52]. Figure 2(a) shows this solution which is characterised by a shock-like structure with $h = 1$ uniform upstream of the moving effective contact line and $h = b$ downstream of it. Surface tension plays an important role near the effective contact line with a pronounced bulge in h, the so-called capillary ridge, and a capillary wave that forms both upstream and downstream of the capillary ridge. The

width of the capillary ridge region scales like $Ca^{1/3}$ and hence becomes much smaller and the slope gets steeper as $Ca \to 0$. This travelling wave solution is used as an exact solution to compare the convergence and accuracy of the numerical solution on an adaptive moving mesh. Figure 2(a) presents the solution $h(x,t)$ at time $t = 10$ obtained using the moving adaptive mesh scheme with $N = 800$ ($\Delta\xi = 0.0075$ = initial Δx and domain length $L = 6$), MMPDE4 (with $\tau = 10^{-2}$) and curvature mesh density function (with $\alpha = \beta = 1$ and $n = 2$). The insets show the increased number of points that are redistributed to the capillary ridge and the capillary wave ahead and behind it. Figure 2(b) shows the curvature mesh density function associated with this solution. The large values of the curvature in the capillary ridge region results in the increased number of points redistributed in this region (see insets in Fig. 2(a)). Figure 2(c) shows the equidistributing coordinate transformation $x = x(\xi, t = 10)$. We observe the large number of points in the capillary ridge region (around $x = 4$) compared to elsewhere (compare also to a uniform mesh represented by the dashed line). The adaptive moving mesh scheme allocates large number of points where there is a large variation in the solution characterized by large variations in its curvature. Figure 3 compares the assumed exact travelling wave solution (solid line) with that computed on a uniform mesh (dashed-dotted line) and adaptive moving mesh (dashed line). The adaptive moving mesh solution is obtained using MMPDE4 (with $\tau = 10^{-2}$) and the curvature mesh density function (with $\alpha = \beta = 1$ and $n = 2$). The solution is translated so that the origin $\xi = 0$ is at the maximum value of h at the capillary ridge. To make a direct comparison between the uniform and adaptive moving mesh schemes, we fix $N = 600$ points for both schemes ($\Delta\xi = 10^{-2}$ =initial Δx and domain length $L = 6$). We illustrate the accuracy of the numerical solution by focussing on the capillary ridge and the capillary wave ahead and behind it (see insets in Fig. 3). We observe that the solution obtained using the moving adaptive mesh scheme is almost identical to the assumed exact solution. The numerical solution using the uniform mesh scheme has not converged to the exact solution for the value of $\Delta\xi = 10^{-2}$ used. The adaptive moving mesh scheme takes an average value of $\Delta x = 2 \times 10^{-5}$ in the capillary ridge and capillary wave regions, hence the almost identical match to the exact solution. However, the uniform mesh scheme requires at least $\Delta\xi \leq 10^{-3}$ ($N = 6000$ points for this case) for this region to be well-resolved (not shown here).

We now consider the error and convergence of the moving adaptive mesh

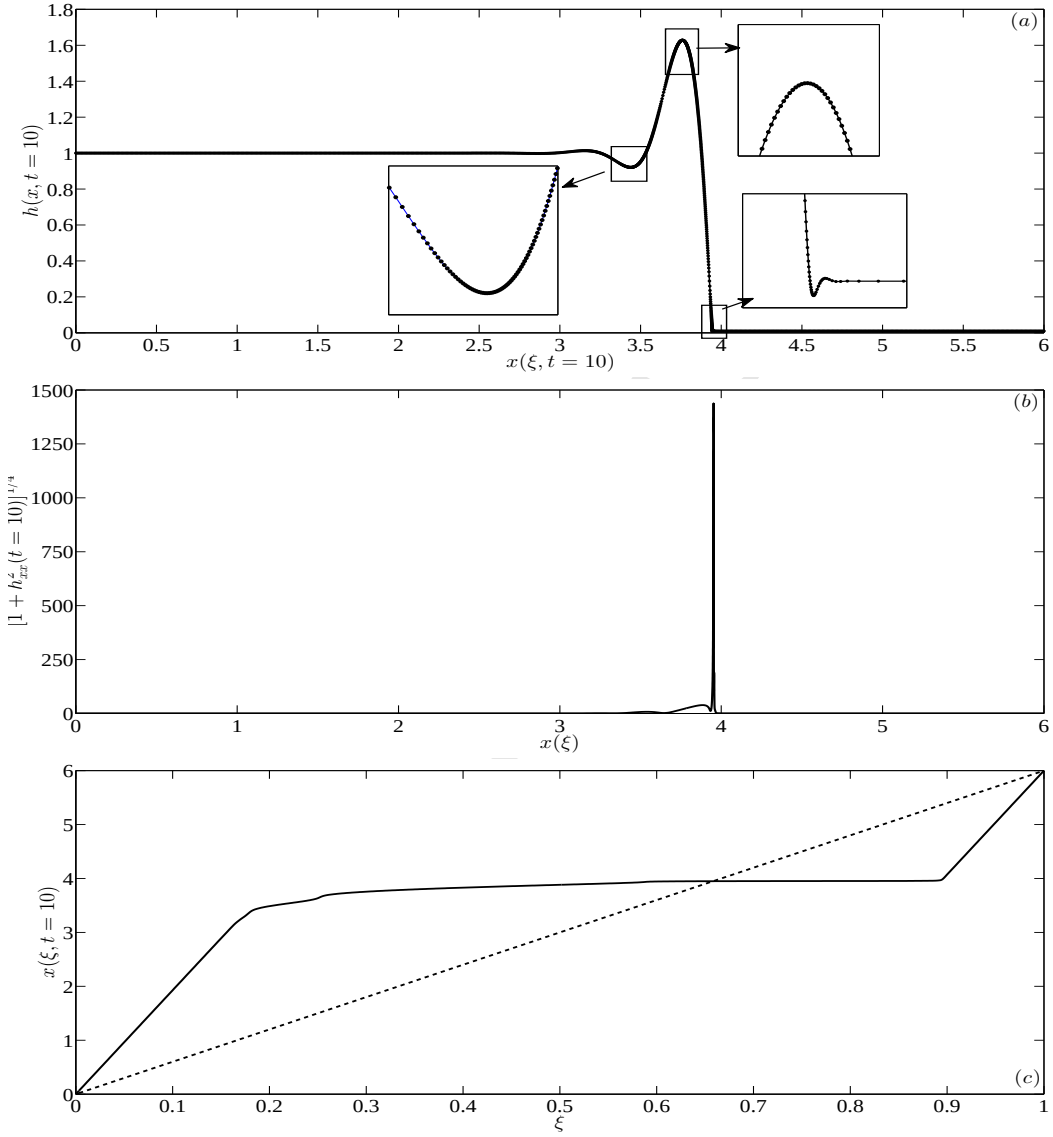

Figure 2: (a) $h(x, t = 10)$, (b) mesh density function at $t = 10$ and (c) $x(\xi, t = 10)$ using MMPDE4 and curvature mesh density function (see text for parameter values). Insets in (a) show the zoomed-in capillary ridge and capillary wave ahead and behind the ridge. The dashed line in (c) represents a uniform mesh.

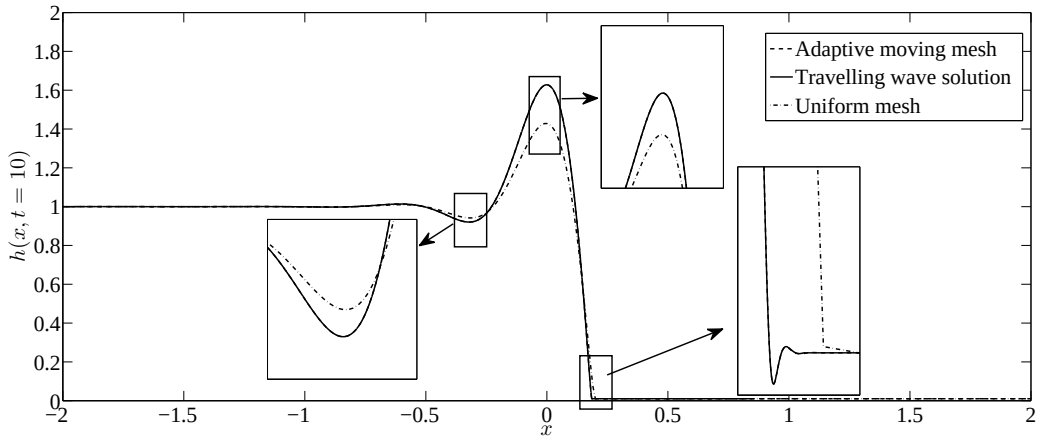

Figure 3: The travelling wave solution for h (solid line) and corresponding numerical results for the uniform (dashed-dotted line) and adaptive moving mesh (dashed line) schemes. Insets show the zoomed-in solution near the capillary ridge and capillary wave ahead and behind the ridge. The adaptive moving mesh solution is obtained using MMPDE4 and the curvature mesh density function (see text for parameter values).

scheme for varying MMPDEs (and their parameters) and mesh density function. Table 1 summarises the error measured in the L_2 norm and CPU time taken to reach $t = 10$ using MMPDEs 4,5,6 and modified MMPDE5 for several values of τ with $N = 1000$ ($\Delta\xi = 6 \times 10^{-3}$ =initial Δx). The curvature mesh density function is used (with $\alpha = \beta = 1$ and $n = 2$). The numerical solution for h used in calculating the error and CPU time is obtained at $t = 10$. We observe from Table 1 that MMPDE4 and MMPDE6 give more accurate solution and take less CPU time compared to MMPDE5 and modified MMPDE5. Four values of the relaxation parameter τ are used with MMPDE4 and MMPDE6 and we observe that the error gets smaller but the CPU time increases as τ decreases. The equations become much stiffer as τ decreases and we were unable to obtain a numerical solution for $\tau < 10^{-4}$ for MMPDE4 and MMPDE6. The same trend holds for MMPDE5 and modified MMPDE5, however, we were unable to obtain a numerical solution for $\tau < 10^{-2}$. It is worth noting that the CPU times for MMPDE5 and modified MMPDE5 are much higher compared to the others, particularly for $\tau = 10^{-1}$. This is due to the equations becoming very stiff for small values of τ. One would need to use their regularised form given by Eqs. (5), (6), which we have not considered here. Therefore, it can be concluded that a value of $\tau = 10^{-3} - 10^{-2}$ is optimal with respect to accuracy and CPU time taken.

MMPDE	N	τ	CPU	Error
	1000	1	41s	3.8×10^{-3}
	1000	10^{-1}	64s	1.8×10^{-4}
4	1000	10^{-2}	71s	3.2×10^{-9}
	1000	10^{-3}	148s	1.0×10^{-8}
	1000	1	48s	2.4×10^{-3}
5	1000	10^{-1}	14650s	5.9×10^{-8}
	1000	1	79s	1×10^{-5}
6	1000	10^{-1}	80s	2.2×10^{-8}
	1000	10^{-2}	130s	1.1×10^{-8}
	1000	10^{-3}	932s	1.1×10^{-8}
	1000	1	116s	2.6×10^{-5}
modified 5	1000	10^{-1}	2088s	1.4×10^{-6}

Table 1: Error (mesaured in the L_2 norm) and CPU time taken to reach $t = 10$ for MMPDEs 4, 5 and 6, and modified MMPDE5 varying the relaxation parameter τ. The numerical solution for h used is obtained at $t = 10$ and the curvature mesh density function is used (see text for parameter values).

Also, MMPDE4 or MMPDE6 are the best in terms of accuracy and CPU time taken (see Table 1). Figure 4 compares the error measured in the L_2 norm for the numerical solutions obtained using the adaptive moving mesh (dashed line using the arc-length mesh density function and dashed-dotted line using the curvature mesh density function) and uniform mesh (solid line) schemes for varying Δx. We choose $\alpha = \beta = 1$ and $n = 2$ for the curvature mesh density function; choosing $n = 4$ gave similar results and we do not report them here. The numerical solution for both schemes used in measuring the error is obtained at $t = 10$ and MMPDE4 is used. We observe that the adaptive moving mesh numerical solution is more accurate with respect to the error compared to the uniform mesh scheme (see approximate slopes shown in Fig. 4 for comparison), i.e., it achieves a higher accuracy for the same number of mesh points. For example, the lowest error recorded is 1.6×10^{-9} for $\Delta x = 5 \times 10^{-3}$ (corresponding to $N = 1200$ points) for the adaptive moving mesh solution using the curvature mesh density function (dashed-dotted line). The error for the uniform mesh solution (solid line) corresponding to this Δx is 2.57×10^{-3}. Moreover, it is also efficient in the number of mesh points used to achieve a desired level of accuracy, i.e., it uses less number of points to achieve the same error as the uniform mesh scheme.

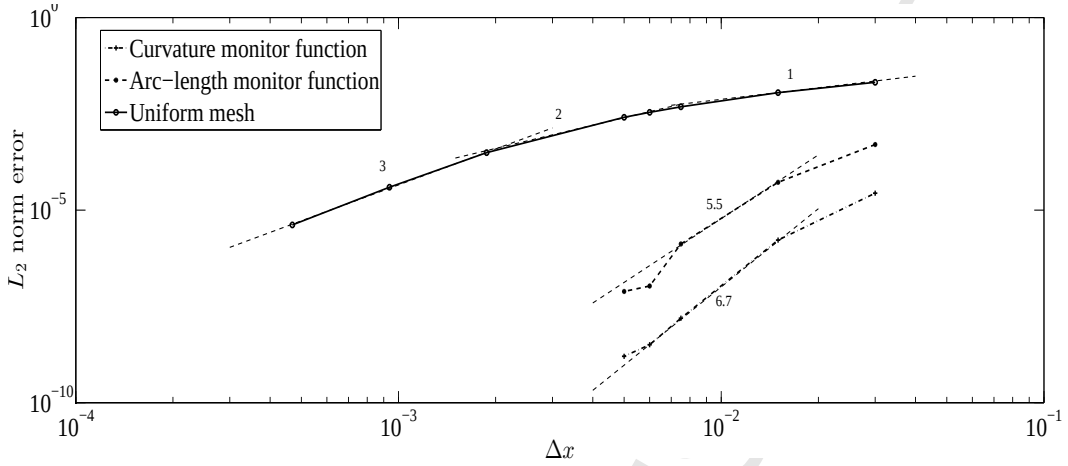

Figure 4: The L_2 norm error for numerical solutions obtained using uniform mesh (solid line) and adaptive moving mesh using the arc-length mesh density function (dashed line) and curvature mesh density function (dashed-dotted line) as a function of Δx. Approximate slopes are shown for comparison between the three solutions. The numerical solution is obtained at $t = 10$ using MMPDE4 (see text for parameter values).

For example, the lowest value of the error recorded for the uniform mesh solution is 4.12×10^{-6} for $\Delta x = 5 \times 10^{-4}$ (corresponding to $N = 12000$ points). The adaptive moving mesh requires $\Delta x \approx 0.02$ (corresponding $N = 300$ points) to record similar error. We also observe that the adaptive moving mesh scheme using the curvature mesh density function (dashed-dotted line) is more accurate and efficient in the number of points used compared to that using the arc-length mesh density function (dashed line). The approximate slopes shown in Fig. 4 confirm this. Table 2 shows the error and CPU time for the uniform mesh and adaptive moving mesh (using the curvature mesh density function with $\alpha = \beta = 1$ and $n = 2$, and MMPDE4 with $\tau = 10^{-2}$) schemes. The numerical solution used in recording the error is obtained at $t = 10$. The error columns summarise Fig. 4 (solid line for uniform mesh scheme and dashed-dotted line for adaptive moving mesh scheme using the curvature mesh density function and MMPDE4). The error for the adaptive moving mesh scheme is much smaller and is achieved using less number of points compared to the uniform mesh scheme. However, with respect to the CPU time taken, the adaptive moving mesh takes more time to reach $t = 10$ compared to the uniform mesh scheme for the same number of points. This is due to the additional adaptive mesh equations that need to be simultaneously solved along with the discretized PDE. Hence, one would need to

N	Error		CPU time taken to $t = 10$	
	Uniform mesh	Adaptive moving mesh	Uniform mesh	Adaptive moving mesh
200	2.08×10^{-2}	2.75×10^{-5}	2s	9s
400	1.11×10^{-2}	1.66×10^{-6}	6.5s	29s
800	4.8×10^{-3}	1.56×10^{-8}	22s	59s
1000	3.4×10^{-3}	3.2×10^{-9}	25s	71s
1200	2.57×10^{-3}	1.6×10^{-9}	43s	119s
3000	3.1×10^{-4}	–	233s	–
6000	3.93×10^{-5}	–	264s	–
12000	4.12×10^{-6}	–	819s	–

Table 2: Comparing the error and CPU time taken to reach $t = 10$ for the uniform mesh and adaptive moving mesh (using the curvature mesh density function and MMPDE4) schemes. The numerical solution is obtained at $t = 10$ (see text for parameter values).

balance the accuracy desired and the CPU time taken to judge the efficacy of the moving adaptive mesh scheme over the uniform mesh scheme. For example, at $N = 800$ there is a threefold increase in time taken by the adaptive moving mesh scheme to reach $t = 10$ (less than a minute, though), however, there is a reduction in the error by five orders of magnitude. For this case, we can certainly conclude that the adaptive moving mesh scheme is more computationally efficient than the uniform mesh scheme. Note that we were unable to compute the solution at $t = 10$ for $N > 1200$. For these values of N the minimum Δx becomes very small and round-off errors dominate resulting in the numerical solution losing stability and becoming unstable.

We now consider the numerical solution of the same PDE but with constant volume boundary conditions (Eq. (15)) and initial condition given by Eq. (17). Informed by the above results, we use the curvature mesh density function (with $\alpha = 1$ and $\beta = 100$ and $n = 2$) and MMPDE4 (with $\tau = 10^{-3}$) for the results shown below. The upstream and downstream domain lengths are fixed at $L_1 = -2$ and $L_2 = 10$, respectively. A uniform mesh is used as the initial condition for MMPDE4. Figure 5(a) shows the time evolution of h using the adaptive moving mesh scheme. In the simulation shown the number of points used is $N = 600$ (corresponding to initial $\Delta x = 0.02$). The time t ranges between 30 to 60. We observe the development of the capillary ridge and capillary wave ahead and behind it near the drop's leading edge

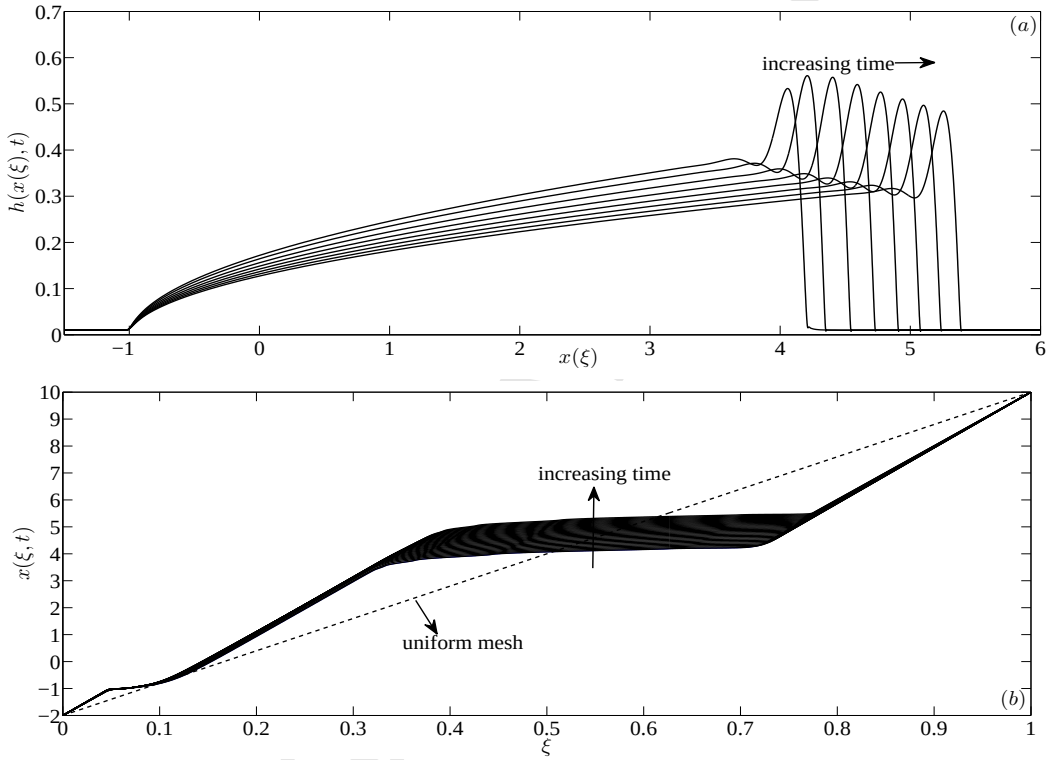

Figure 5: Time evolution ($t = 30-60$) of (a) $h(x,t)$ and (b) $x(\xi,t)$, subject to the boundary conditions Eq. (15) using MMPDE4 and curvature mesh density function (see text for parameter values). The dashed line shows the initial uniform mesh.

where it connects onto the precursor film (the wave ahead of the ridge has a much smaller width and is not seen on the scale shown). This is similar to the previous constant flux case but here the drop height steadily decreases in time as it spreads down the inclined substrate. We also observe a capillary wave that develops near the trailing edge of the drop. This is of similar width to the one near the leading edge and is also not seen on the scale shown. Figure 5(b) shows the corresponding equidistributing coordinate transformation $x(\xi, t)$. The dashed line shows the initial uniform mesh. We observe from figure 5(b) that majority of mesh points are redistributed to the capillary ridge and waves near the leading edge of the drop where changes in curvature are large. There is also redistribution of points near the trailing edge where there are also changes in curvature, although not as large as near the leading edge. The redistribution of points is observed to be more biased towards the leading edge of the front due to large changes in curvature there. To better control this redistribution so that the solution near trailing edge is also accurately resolved, we modify the parameters in the curvature mesh density function as follows:

$$\rho(x,t) = \sqrt{\alpha + \beta(x) h_{xx}^2}, \tag{36}$$

where

$$\beta(x) = \begin{cases} \beta_0, & \text{if } L_1 \le x \le 1, \\ \beta_1, & \text{if } 1 < x \le L_2, \end{cases} \tag{37}$$

$\beta_0 \gg \beta_1$ in the interval $[L_1, 1]$ ensures that more points are redistributed to the trailing edge region (compared to the previous simulation) while the low value of β_1 in the interval $[1, L_2]$ reduces the redistribution to the leading edge even though there are large changes in curvature there. We note that the choice of $x = 1$ is arbitrary in Eq. (37); any location at the upstream end of the trailing edge capillary wave would suffice. The piecewise constant values β_0 and β_1 are based on some *a priori* knowledge of the values of the solution curvature at the leading and trailing edges of the drop. Figure 6(a) shows the time evolution using the curvature-based mesh density function given by Eq. (36) with $\alpha = 1$, $\beta_0 = 3$ and $\beta_1 = 10^{-3}$. In the simulation shown the number of points used is $N = 600$ (corresponding to initial $\Delta x = 0.02$). The time t ranges between 30 to 60. The solution structure, at least visually, looks more accurate than that in Fig. 5(a). Figure 6(b) shows the corresponding equidistributing coordinate transformation $x(\xi, t)$. The dashed line shows the initial uniform mesh. We clearly observe a marked change in the redistribution of points towards the trailing and leading edges

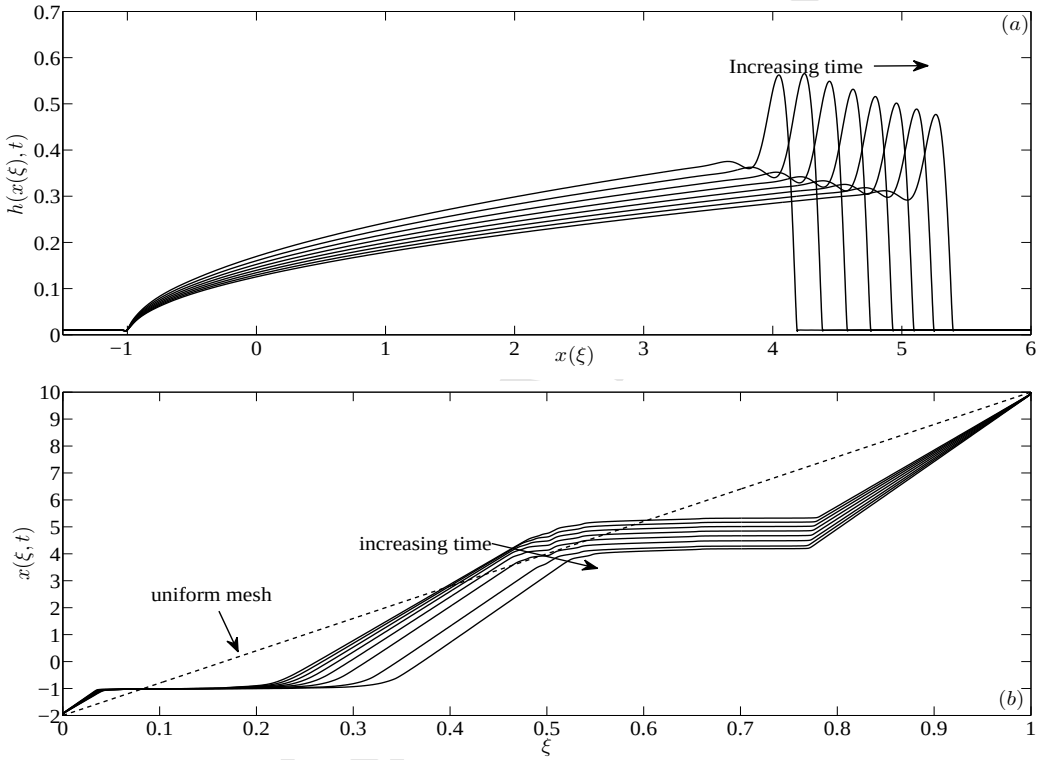

Figure 6: (a) Time evolution $(t = 30 - 60)$ of (a) $h(x,t)$ and (b) $x(\xi,t)$, using MMPDE4 and the curvature mesh density function given by Eq. (36) (see text for parameter values). The dashed line in (b) shows the initial uniform mesh.

in comparison to that shown in Fig. 5(b). The effect of this on the accuracy of the numerical solution is shown in Fig. 7(a, b), which plots the numerical solution at $t = 60$ for the adaptive moving mesh using the curvature mesh density function given by Eq. (36) with $\alpha = 1$, $\beta_0 = 3$ and $\beta_1 = 10^{-3}$, and Eq. (34) with $\alpha = 1$ and $\beta = 100$, respectively. Inspection of the figures (see zoomed-in insets at leading and trailing edge of the drop), clearly show the well-resolved capillary wave near the trailing edge in (a) compared to that in (b); the leading edge structures are similar in both.

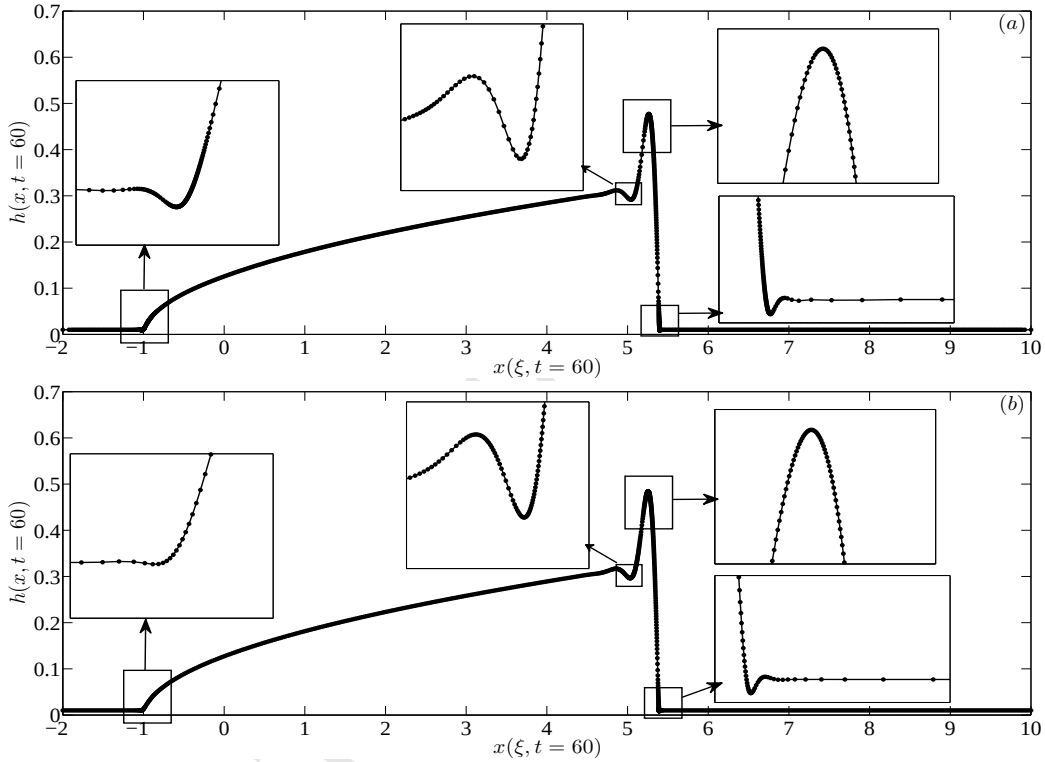

Figure 7: The numerical solution at $t = 60$ using the mesh density function given by (a) Eq. (36) and (b) Eq. (34) (see text for parameter values).

We now include the effect of surfactants into the problem. We first consider the problem with no flux boundary conditions (Eq. 14). Levy and co-workers [22, 23] have analysed the travelling wave structure of this problem and have shown the existence of multiple waves travelling with the same speed. In principle, one can use this multiple travelling wave structure to compare

the convergence and accuracy of the numerical solution. We do not do this here but as an alternative use the numerical solution using a uniform mesh with sufficiently small Δx as an exact solution to compare with that using an adaptive mesh. We note here that the mesh density functions described in §2 are based on the solution having only a single component, i.e., only h. When the solution has multi-components (as in the case here where the solution has two components h and Γ), these mesh density functions need to be adapted to account for variations in the other components (which need not necessarily be aligned to variations in one component). Therefore, the curvature mesh density function in Eq. (34) is modified by the addition of the curvature in Γ and discretized using finite differences as:

$$\text{modified curvature}: \quad \hat{\rho}(x,t) = (1 + \beta h_{xx}^2 + \omega \Gamma_{xx}^2)^{\frac{1}{2}}, \tag{38}$$

$$\text{semi-discretisation}: \quad \hat{\rho}(x_j,t) = (1 + \beta h_{xx,j}^2 + \omega \Gamma_{xx,j}^2)^{\frac{1}{2}}, \tag{39}$$

where the second order derivatives are discretized as in Eq. (34). Here β and ω are user-specified parameters. We also used the arc-length Γ_x instead of it's curvature in Eq. (38) but the redistribution of mesh points were more sensitive to changes in the curvature of Γ rather than it's arc-length. We use the same smoothing process for the discretized mesh density function as described in §4 using Eq. (35). Figures $8(a,b)$ show the time evolution of $h(x,t)$ and $\Gamma(x,t)$ respectively, to a travelling wave solution using the adaptive moving mesh scheme with $N = 1000$ points corresponding to initial mesh size $\Delta x = 0.04$ and domain length $L = 40$. The results are shown for t ranging between 0 to 90. We use the curvature mesh density function given by Eq. (38) (with $\alpha = 1$ and $\beta = \omega = 10^5$) and MMPDE4 (with $\tau = 10^{-3}$). The solution for h shows a multiple travelling wave structure characterized by a capillary ridge and capillary waves near it's leading edge. In addition, a "step"-like structure where $h \approx 2b$ (twice the precursor film thickness) develops ahead of this leading edge [13, 20, 22, 23, 46, 47]. The solution for Γ, although continuous, displays large variation in it's gradient, particularly, near it's leading edge. Figures $9(a,b)$ present the solution $h(x,t)$ and $\Gamma(x,t)$, respectively, at time $t = 90$. The insets show the increased number of points that are redistributed near the capillary ridge and capillary waves ahead and behind it, and the step-like structure (Fig. $9(a)$) which are controlled by the curvature in h component of the mesh density function. In addition, we see redistribution of points around the maximum in Γ and it's leading

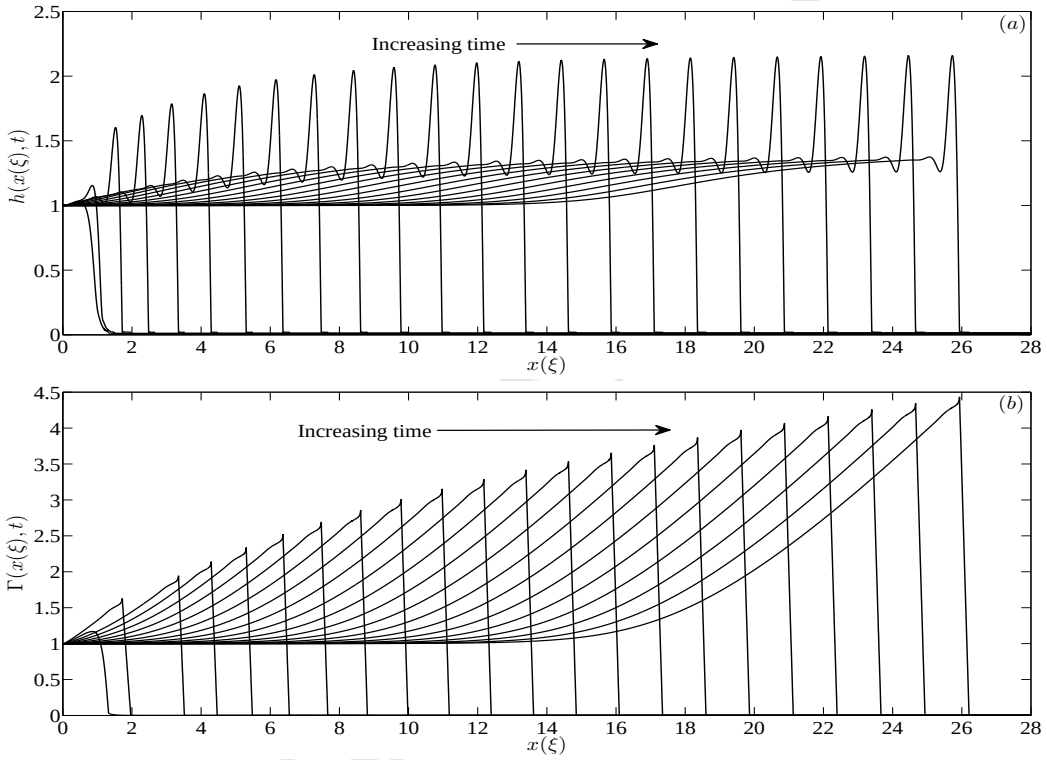

Figure 8: Time evolution ($t = 0-90$) of (a) $h(x,t)$ and (b) $\Gamma(x,t)$, subject to the boundary conditions Eq. (14) using MMPDE4 and a modified curvature mesh density function given by Eq. (38). See text for parameter values.

edge (Fig. 9(b)) controlled by the curvature in Γ component of the mesh density function. In Figs. 10(a, b) we compare the numerical solution for

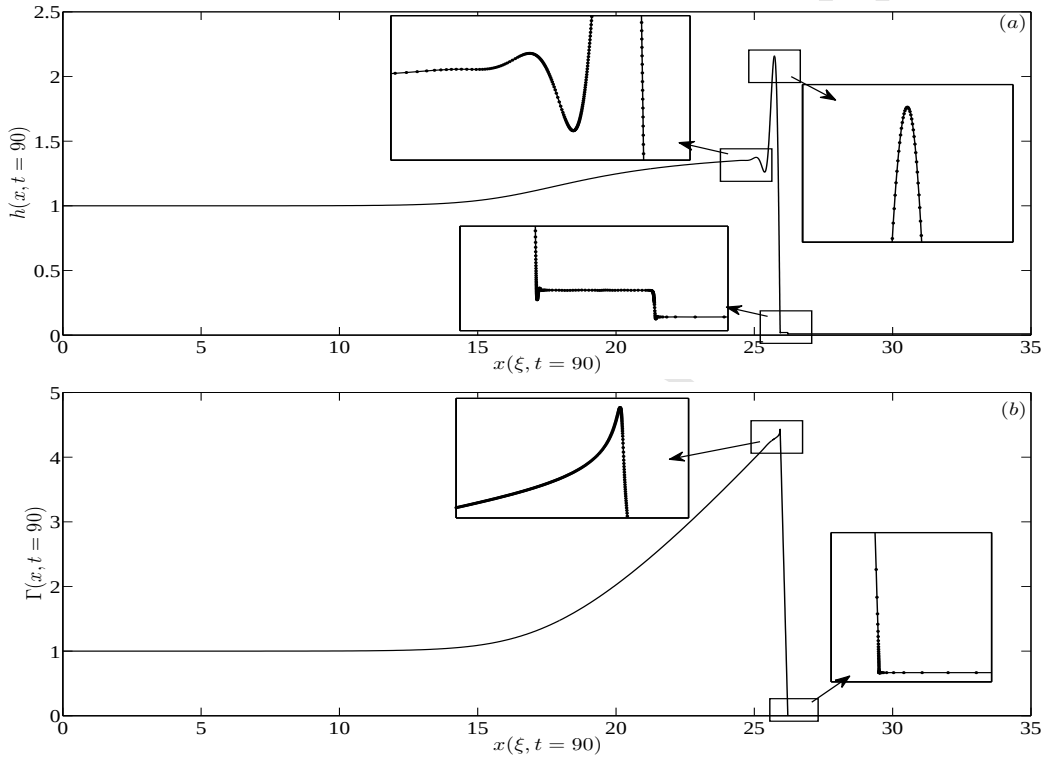

Figure 9: (a) $h(x, t = 90)$ and (b) $\Gamma(x, t = 90)$. Insets in (a, b) show the zoomed-in wave-like structures in h and steep gradient in Γ around the location of it's maximum and leading edge. See text for parameter values.

h and Γ, respectively, at $t = 90$, using a uniform mesh (with $N = 10^3$ or $\Delta x = 0.04$; $N = 4 \times 10^4$ or $\Delta x = 10^{-3}$; $N = 4 \times 10^5$ or $\Delta x = 10^{-4}$) and an adaptive moving mesh (with $N = 1000$ or initial $\Delta x = 0.04$). We observe that the adaptive moving mesh numerical solution is almost indistinguishable from the uniform mesh numerical solution corresponding to $\Delta x = 10^{-3}, 10^{-4}$. Moreover, the insets in figures 10(a, b) show that there is still an error (albeit small) between the uniform mesh solution for $\Delta x = 10^{-3}, 10^{-4}$, particularly at the capillary ridge in h and the maximum in Γ. This implies that the accuracy and convergence of the uniform mesh scheme is restricted to very small values of Δx in these regions. In contrast, the adaptive moving mesh scheme by redistributing more points to these regions (starting from a modest

number of points, $N = 1000$) can very effectively control the accuracy and convergence of the numerical solution there.

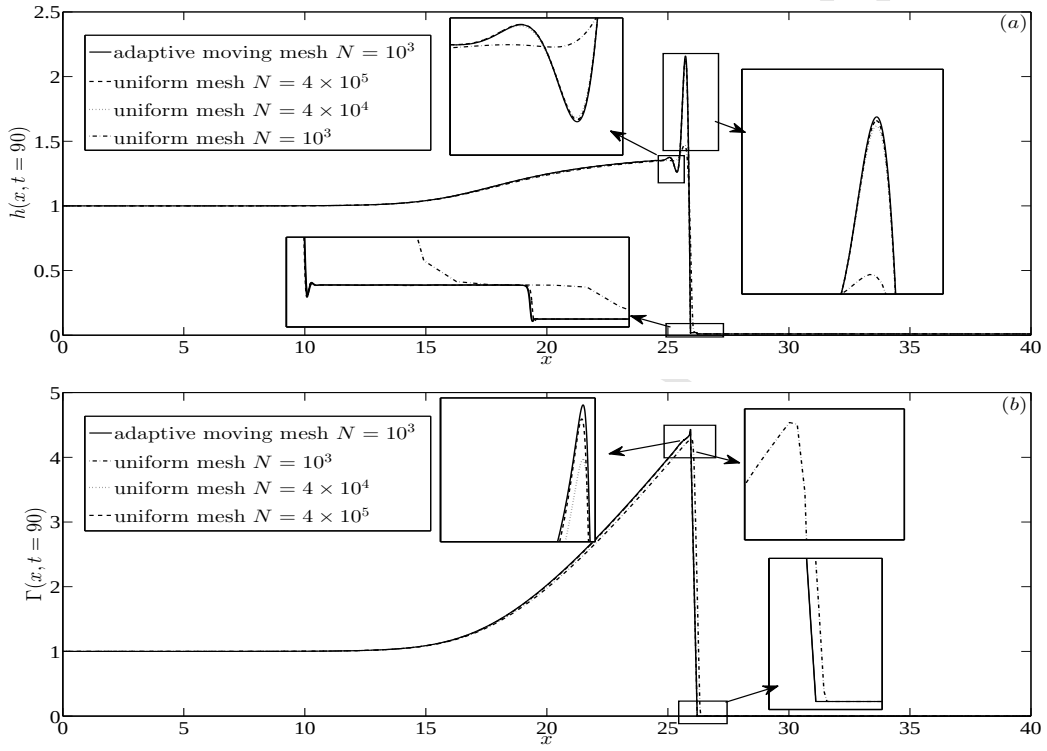

Figure 10: A comparison of the numerical solution of (a) $h(x, t = 90)$ and (b) $\Gamma(x, t = 90)$ using a uniform mesh (with $N = 10^3$ or $\Delta x = 0.04$; $N = 4\times10^4$ or $\Delta x = 10^{-3}$; $N = 4\times10^5$ or $\Delta x = 10^{-4}$) and an adaptive moving mesh (with $N = 1000$ or initial $\Delta x = 0.04$). The insets show the zoomed-in solution at locations where there is large variation in h and Γ. See text for parameter values.

We now consider the numerical solution of the same PDEs but with constant volume boundary conditions. The upstream and downstream domain lengths are fixed at $L_1 = -3$ and $L_2 = 17$, respectively. Figs. $11(a, b, c)$ show the time evolution of $h(x, t)$, $\Gamma(x, t)$ and $x(\xi, t)$, respectively, using an adaptive moving mesh with $N = 1500$ points corresponding to initial mesh size $\Delta\xi = 0.013$. The results are shown for t ranging between 0 to 400. We use the curvature mesh density function given by Eq. (38) (with $\alpha = 1$ and $\beta = \omega = 10^5$) and MMPDE4 (with $\tau = 10^{-3}$). In addition to the wave-like structures observed at the leading edge of the drop (similar to the constant

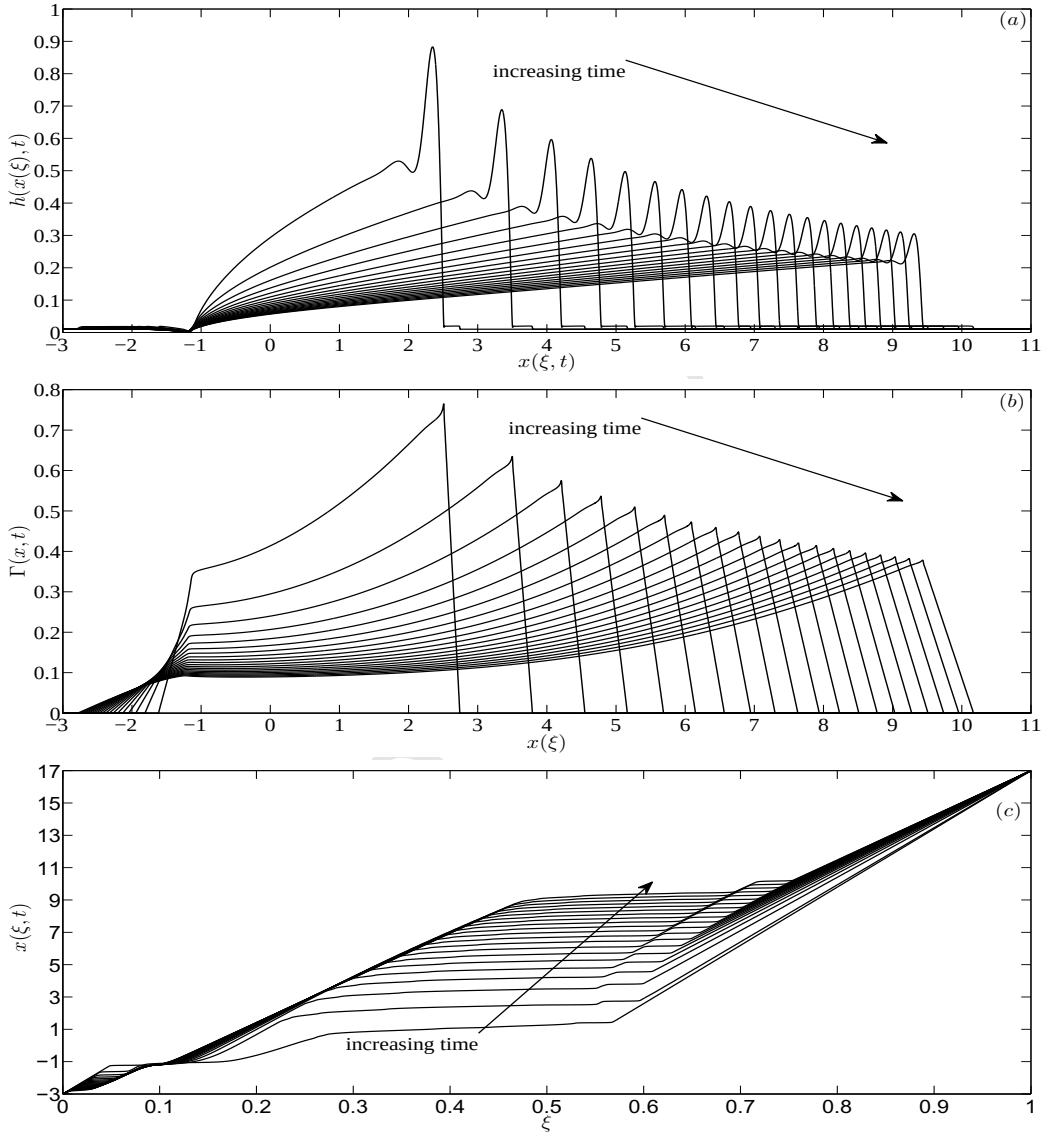

Figure 11: Time evolution ($t = 0 - 400$) of (a) $h(x,t)$, (b) $\Gamma(x,t)$ and (c) $x = x(\xi,t)$.

flux case above), there is now also a front in h travelling upstream of the trailing edge of the drop (see Fig. 11(a)). There is also a steep gradient in Γ near the trailing edge of the drop (see Fig. 11(b)) in addition to the one near the drop's leading edge. The corresponding equidistributing coordinate transformation $x(\xi, t)$ in Fig. 11(c) shows that the mesh points are redistributed with more points in the region near the leading edge and trailing edges ($\Delta x \approx 10^{-4}$), where there is a large variation in both h and Γ, and fewer points elsewhere ($\Delta x \approx 0.1$); note the initial $\Delta x = 0.013$ uniform everywhere. Figures 12(a, b) present the solution $h(x, t)$ and $\Gamma(x, t)$, respectively, at time $t = 400$. The insets show the increased number of points that are redistributed near the capillary ridge and capillary waves ahead and behind it, the step-like structure ahead of the drop's leading edge and the front upstream of the drop's trailing edge (see Fig. 12(a)) which are controlled by the curvature in h component of the mesh density function. In addition, we see redistribution of points around the maximum in Γ and it's leading edge and trailing edge (see Fig. 12(b)) controlled by the curvature in Γ component of the mesh density function. In Figures 13(a, b) we compare the numerical solution for h and Γ, respectively, at $t = 400$, using a uniform mesh (with $N = 1500$ or $\Delta x = 0.013$; $N = 2 \times 10^4$ or $\Delta x = 10^{-3}$) and an adaptive moving mesh (with $N = 1500$ or $\Delta\xi = 0.013 =$ initial Δx). We observe that the adaptive moving mesh numerical solution is almost indistinguishable from the uniform mesh numerical solution corresponding to $\Delta x = 10^{-3}$. However, on closer inspection there are discernible differences between the two solutions for both h and Γ, particularly, near the leading and trailing edges of the drop (see solid and dashed lines in the insets in Figs. 13(a, b)). We believe that the uniform mesh solution at $\Delta x = 10^{-3}$ has not fully converged in these regions in comparison to the adaptive mesh solution where $\Delta x \approx 10^{-4}$ in these regions. We would need to use a smaller value of Δx to confirm this which we have not done here. This again highlights the usefulness of the adaptive moving mesh scheme in redistributing more points to these regions (starting from a modest number of points, $N = 1500$) enabling to effectively control the accuracy and convergence of the numerical solution there.

6. Conclusions

We have successfully applied the r-adaptive moving mesh method based on MMPDEs and mesh density functions to two prototype one-dimensional thin

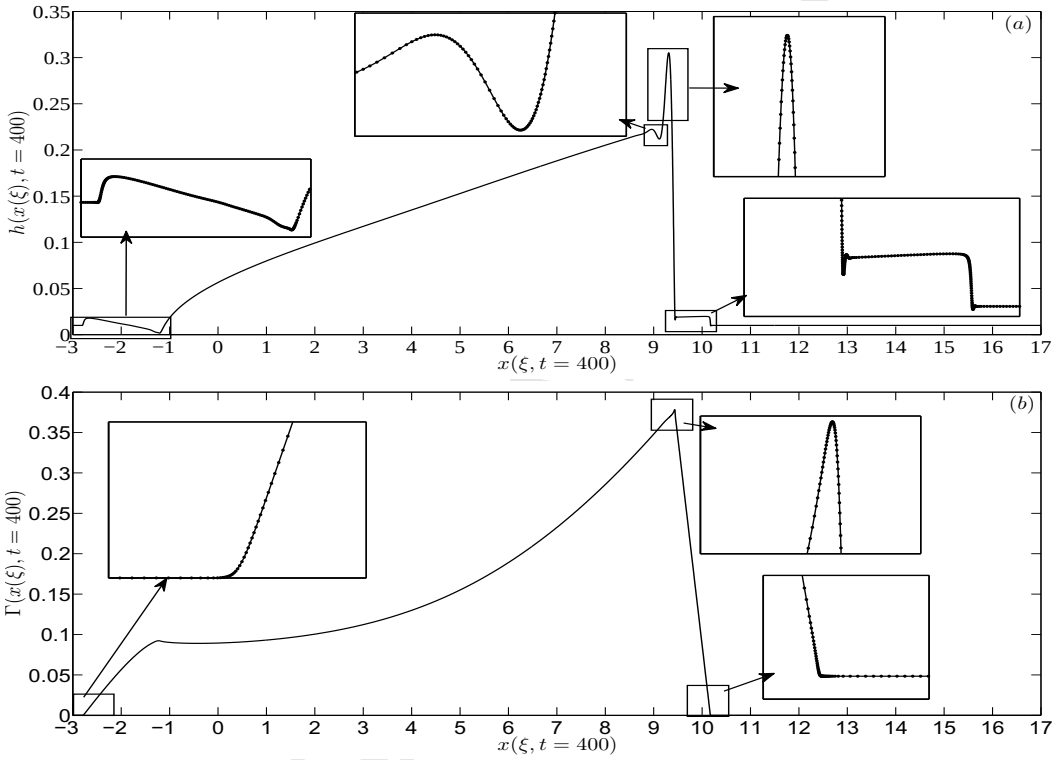

Figure 12: (a) $h(x, t = 400)$ and (b) $\Gamma(x, t = 400)$. Insets in (a, b) show the zoomed-in wave-like structures in h and steep gradient in Γ around the drop's leading and trailing edges. See text for parameter values.

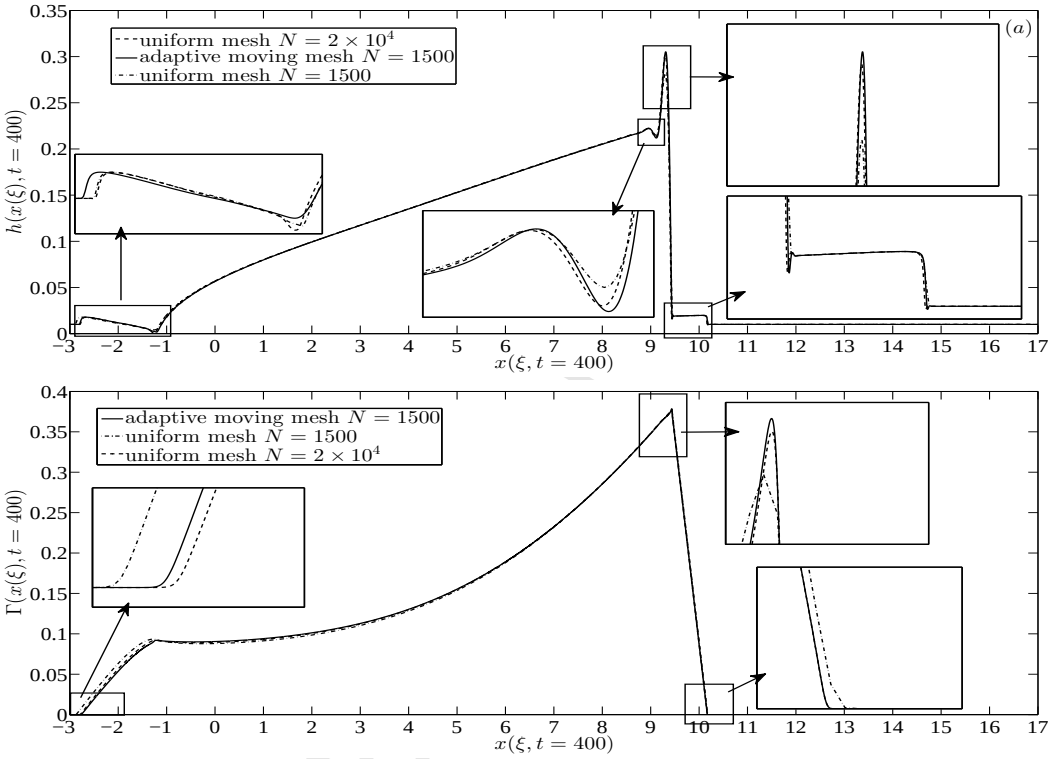

Figure 13: A comparison of the numerical solution of (a) $h(x, t = 400)$ and (b) $\Gamma(x, t = 400)$ using a uniform mesh (with $N = 1500$ or $\Delta x = 0.013$; $N = 2 \times 10^4$ or $\Delta x = 10^{-3}$) and an adaptive moving mesh ($N = 1500$ or $\Delta\xi = 0.013 =$ initial Δx). The insets show the zoomed-in solution at locations where there is large variation in h and Γ. See text for parameter values.

film equations represented by a coupled system of higher order parabolic PDEs. The main highlight of the results are shown in Fig. 4 and Table 2 which enable direct comparison with the uniform mesh scheme. We observe that the error for a fixed number of mesh points is always much lower for the adaptive moving mesh schemes compared to the corresponding uniform mesh scheme. Alternatively, for a prescribed error, the adaptive moving mesh scheme achieves this with far less number of points compared to the uniform mesh scheme. However, the adaptive moving mesh scheme takes much longer CPU times than the corresponding uniform mesh scheme due to the extra mesh PDE that needs to be solved along with the underlying PDEs. This difference between CPU times is not that large if the desired error is not too small.

An appropriate choice of the mesh density function is shown to be crucial to the accuracy of the adaptive mesh scheme. We have found the curvature mesh density function to be most appropriate for the thin film problems considered here. A significant result is related to adaptation of this mesh density function to accurately resolve the solution at multiple locations. The simplest option using a piecewise constant weight parameter allowed the mesh points to be redistributed accordingly based on some *a priori* knowledge of the solution (see Figs. 6, 7). Another option could be to use a mixed mesh density function, i.e., combination of curvature and arc-length mesh density functions. These would need to be explored in future. This also highlights one of the drawbacks of r-adaptive methods which redistribute a fixed number of points in contrast to hp-adaptive methods which allow for dynamic allocation of mesh points. One would need to start with a large number of points if there are multiple locations to resolve along with a judiciously chosen mesh density function. We have also adapted the curvature mesh density function to include multiple solution components (based on Huang & Russell [37]). This enables us to accurately resolve the complicated multiple structures in both solution components (which need not necessarily be aligned with each other) compared to the uniform mesh scheme (see Figs. 10, 13).

In conclusion, our results indicate great promise in terms of simplicity in its implementation and efficiency (in comparison to uniform mesh schemes) for MMPDEs-based moving adaptive mesh methods to be applied on a regular basis in thin film flow problems. Although we have only considered a specific form of the underlying PDEs, the general framework presented in this paper can be utilised for a thin-film equation coupled to other PDEs

such as, electric field, temperature (in non-isothermal problems), etc., of the form considered here. Moreover, this framework can be easily extended to two-dimensional thin film flow problems, this is currently being undertaken and will be reported elsewhere. However, there are challenging problems in the thin film literature (for example, those involving moving contact lines - analogous to the case when the precursor film thickness $b \ll 1$) where this method needs to be further tested before its success can be guaranteed.

Acknowledgement

This work was a part of Abdulghani Alharbi's PhD research at Keele University. Abdulghani gratefully acknowledges financial support by the government of the Kingdom of Saudi Arabia and Taibah University.

Appendix A. A *non-conservative* finite difference semi-discretisation scheme

We can write a *non-conservative* semi-discretisation scheme for the spatial derivatives in Eqs. (12),(13). To do this, we write Eqs. (12),(13) in the following form:

$$h_t + Ca\left[\frac{h^3}{3}p_x\right]_x - \beta_1\left[\frac{h^3}{3}h_x\right]_x - \left[\frac{h^2}{2}\Gamma_x\right]_x + \alpha_1\left[\frac{h^3}{3}\right]_x = 0, \tag{A.1}$$

$$\Gamma_t + Ca\left[\frac{h^2\Gamma}{2}p_x\right]_x - \beta_1\left[\frac{h^2\Gamma}{2}h_x\right]_x - [(h\Gamma+\delta)\Gamma_x]_x + \alpha_1\left[\frac{h^2\Gamma}{2}\right]_x = 0, \tag{A.2}$$

$$p = h_{xx}. \tag{A.3}$$

A *non-conservative* semi-discretization scheme for a function of the form $[au_x]_x$ on a uniform mesh $\mathcal{J}_u^c$ (say) using finite differences can be written as:

$$[au_x]_{x,j} = \frac{1}{x_{\xi,j}}\left[\frac{au_\xi}{x_\xi}\right]_{\xi,j} = \frac{1}{x_{\xi,j}}\left[\frac{[au_\xi]_{\xi,j}}{x_{\xi,j}} - \frac{1}{x_{\xi,j}^2}a_j u_{\xi,j} x_{\xi\xi,j}\right] = \left(\frac{2}{\Delta x_j + \Delta x_{j-1}}\right)^2 \left[a_{j+\frac{1}{2}}\Delta u_j - a_{j-\frac{1}{2}}\Delta u_{j-1}\right] - a_j\left(\frac{\Delta u_j + \Delta u_{j-1}}{\Delta x_j + \Delta x_{j-1}}\right)\left(\frac{\Delta x_j - \Delta x_{j-1}}{\Delta x_j \Delta x_{j-1}}\right). \tag{A.4}$$

Using the above, a *non-conservative* semi-discretisation scheme for the spatial derivatives in Eqs. (A.1)-(A.3) on the uniform mesh $\mathcal{J}_{h,\Gamma}^c$ using finite differences can be written as, keeping the time derivative continuous,

$$h_{t,j} + \left[Ca\frac{h^3}{3}p_x - \beta_1\frac{h^3}{3}h_x - \frac{h^2}{2}\Gamma_x + \alpha_1\frac{h^3}{3}\right]_{x,j} = 0, \quad j = 2,\ldots,N, \tag{A.5}$$

$$\Gamma_{t,j} + \left[Ca\frac{h^2\Gamma}{2}p_x - \beta_1\frac{h^2\Gamma}{2}h_x - (h\Gamma + \delta)\Gamma_x + \alpha_1\frac{h^2\Gamma}{2}\right]_{x,j} = 0, \quad j = 2,\ldots,N, \tag{A.6}$$

$$p_j = h_{xx,j} = \left(\frac{2}{\Delta x_j + \Delta x_{j-1}}\right)^2 [\Delta h_j - \Delta h_{j-1}] - \left(\frac{\Delta h_j + \Delta h_{j-1}}{\Delta x_j + \Delta x_{j-1}}\right)\left(\frac{\Delta x_j - \Delta x_{j-1}}{\Delta x_j \Delta x_{j-1}}\right), \quad j = 2,\ldots,N. \tag{A.7}$$

The spatial derivatives appearing in Eqs. (A.5)-(A.7)are discretised as follows:

$$[ap_x]_{x,j} = \left(\frac{2}{\Delta x_j - \Delta x_{j-1}}\right)^2 \left[a_{j+\frac{1}{2}}\Delta p_j - a_{j-\frac{1}{2}}\Delta p_{j-1}\right] - a_j \left(\frac{\Delta p_j - \Delta p_{j-1}}{\Delta x_j - \Delta x_{j-1}}\right)\left(\frac{\Delta x_j - \Delta x_{j-1}}{\Delta x_j \Delta x_{j-1}}\right), \tag{A.8}$$

$$[ah_x]_{x,j} = \left(\frac{2}{\Delta x_j - \Delta x_{j-1}}\right)^2 \left[a_{j+\frac{1}{2}}\Delta h_j - a_{j-\frac{1}{2}}\Delta h_{j-1}\right] - a_j \left(\frac{\Delta h_j - \Delta h_{j-1}}{\Delta x_j - \Delta x_{j-1}}\right)\left(\frac{\Delta x_j - \Delta x_{j-1}}{\Delta x_j \Delta x_{j-1}}\right), \tag{A.9}$$

$$[a\Gamma_x]_{x,j} = \left(\frac{2}{\Delta x_j - \Delta x_{j-1}}\right)^2 \left[a_{j+\frac{1}{2}}\Delta \Gamma_j - a_{j-\frac{1}{2}}\Delta \Gamma_{j-1}\right] - a_j \left(\frac{\Delta \Gamma_j - \Delta \Gamma_{j-1}}{\Delta x_j - \Delta x_{j-1}}\right)\left(\frac{\Delta x_j - \Delta x_{j-1}}{\Delta x_j \Delta x_{j-1}}\right), \tag{A.10}$$

$$[h^3]_x = 2\left(\frac{h^3_{j+1/2} - h^3_{j-1/2}}{x_{j+1} - x_{j-1}}\right). \tag{A.11}$$

Approximations to terms of the form $a_{j+1/2}$ and $a_{j-1/2}$ are the same as defined previously. This scheme also requires fictitious points $(h, x)_0$ and $(h, x)_{N+2}$ which are the same as described previously.

References

[1] T. G. Myers, Surface tension driven thin film flows, in: The Mechanics of Thin Film Coatings, Wiley, 1996.

[2] T. G. Myers, Thin films with high surface tension, SIAM Rev. 40 (3) (1998) 441–462.

[3] J. B. Grotberg, Respiratory fluid mechanics and transport processes, Annu. Rev. Biomed. Eng. 3 (2001) 421–457.

[4] R. Braun, Dynamics of the tear film, Annu. Rev. Fluid Mech. 44 (2012) 267–297.

[5] R. Griffiths, The dynamics of lava flows, Annu. Rev. Fluid Mech. 32 (2000) 477–518.

[6] A. Oron, S. H. Davis, S. G. Bankoff, Long-scale evolution of thin liquid films, Rev. Mod. Phy. 69 (1997) 931–980.

[7] R. Craster, O. Matar, Dynamics and stability of thin liquid films, Rev. Mod. Phys. 81 (2009) 1131–1198.

[8] A. L. Bertozzi, The mathematics of moving contact lines in thin liquid films, Notices Amer. Math. Soc. 45 (6) (1998) 689 – 697.

[9] S. Troian, E. Herbolzheimer, S. Safran, Model for the fingering instability of the spreading surfactant drops, Phys. Rev. Lett. 65 (1990) 333–336.

[10] A. L. Bertozzi, M. P. Brenner, Linear stability and transient growth in driven contact lines, Phys. Fluids 9 (1997) 530–539.

[11] L. Kondic, Instabilities in gravity driven flow of thin fluid films, SIAM Rev. 45 (1) (2003) 95–115.

[12] M. R. E. Warner, R. V. Craster, O. K. Matar, Fingering phenomena associated with insoluble surfactant spreading on thin liquid films, Fluid Mech. 510 (2004) 169–200.

[13] B. Edmonstone, O. Matar, R. Craster, Surfactant-induced fingering phenomena in thin film flow down an inclined plane, Physica D: Nonlinear Phenomena 209 (2005) 62–79.

[14] O. E. Jensen, S. Naire, The spreading and stability of a surfactant-laden drop on a prewetted substrate, J. Fluid Mech. 554 (2006) 5–24.

[15] J. V. Goddard, S. Naire, The spreading and stability of a surfactant-laden drop on an inclined prewetted substrate, J. Fluid Mech. 772 (2015) 535–568.

[16] F. Bernis, Viscous flows, fourth order nonlinear degenerate parabolic equations and singular elliptic problems, in: Free boundary problems: theory and application, Vol. 323, Pitman Research Notes in Mathematics, 1995, pp. 40–56.

[17] L. Kondic, J. Diez, Pattern formation in the flow of thin films down an incline: Constant flux configuration, J. Comp. Phys. 13 (11) (2001) 3168–3184.

[18] J. A. Diez, L. Kondic, Computing three-dimensional thin film flows including contact lines, Journal of Computational Physics 183 (2002) 274–306.

[19] M. R. E. Warner, R. V. Craster, O. K. Matar, Fingering phenomena created by a soluble surfactant deposition on a thin liquid film, Phys. Fluids 16 (2004) 2933–2951.

[20] B. D. Edmonstone, O. K. Matar, R. V. Craster, Flow of surfactant-laden thin films down an inclined plane, J. Engrg. Math. 50 (2004) 141–156.

[21] B. Edmonstone, R. Craster, O. Matar, Surfactant-induced fingering phenomena beyond the critical micelle concentration, Fluid Mech. 564 (2006) 105–138.

[22] R. Levy, M. Shearer, The motion of a thin liquid film driven by surfactant and gravity., SIAM J. Appl. Math. 66 (5) (2006) 1588–1609.

[23] R. Levy, M. Shearer, T. P. Witelski, Gravity-driven thin liquid films with insoluble surfactant: smooth traveling waves, Eur. J. Appl. Math. 18 (2007) 679–708.

[24] A. Mavromoustaki, O. Matar, R. Craster, Dynamics of a climbing surfactant-laden film II: Stability, J. Colloid Interface Sci. 371 (2012) 121–135.

[25] J. Barrett, J. Blowey, H. Garcke, Finite element approximation of a fourth order degenerate parabolic equation, Numer. Math. 80 (1998) 525–556.

[26] G. Grün, M. Rumpf, Nonnegativity preserving convergent schemes for the thin film equation, Numer. Math. 87 (2000) 113–152.

[27] A. Heryudono, R. Braun, T. Driscoll, K. Maki, L. Cook, P. King-Smith, Single-equation models for the tear film in a blink cycle: realistic lid motion, Math. Med. Biol. 4 (2007) 347–377.

[28] P. Keast, P. H. Muir, Algorithm 688: EPDCOL: A more efficient PDECOL code, ACM Trans. Math. Softw. 17 (2) (1991) 153–166.

[29] J. Verwer, J. Blom, J. M. Sanz-Serna, An adaptive moving grid method for one-dimensional systems of partial differential equations, J. Comp. Phys. 82 (1989) 454–486.

[30] J. G. Verwer, J. G. Blom, R. M. Furzeland, P. A. Zegeling, A moving grid method for one-dimensional PDEs based on the method of lines, SIAM Philadelphia, 1989, Ch. 12, pp. 160–175.

[31] R. Furzeland, J. Verwer, P. A. Zegeling, A numerical study of three moving grid methods for one-dimensional partial differential equations which are based on the method of lines, J. Comp. Phys. 89 (1990) 349–388.

[32] J. Blom, P. Zegeling, Algorithm 731: A moving-grid interface for systems of one-dimensional partial differential equations, ACM Trans. Math. Softw. 20 (1994) 194–214.

[33] P. Sun, R. D. Russell, J. Xu, A new adaptive local mesh refinement algorithm and its application on fourth order thin film flow problem, J. Comp. Phys. 224 (2007) 1021–1048.

[34] Y. Li, D. Jeong, J. Kim, Adaptive mesh refinement for simulation of thin film flows, Meccanica 49 (2013) 239–252.

[35] Y. Lee, H. M. Thompson, P. H. Gaskell, An efficient adaptive multigrid algorithm for predicting thin film flow on surfaces containing localised topographic features, Comp. Fluids 37 (2007) 838855.

[36] Y. Lee, H. M. Thompson, P. H. Gaskell, The efficient and accurate solution of continuous thin film flow over surface patterning and past occlusions, Int. J. Num. Meth. Fluids 56 (2008) 13751381.

[37] W. Huang, R. D. Russell, *Adaptive Moving Mesh Methods*, Springer, 2011.

[38] C. J. Budd, W. Huang, R. D. Russell, Adaptivity with moving grids, Acta Numerica 18 (2009) 111–241.

[39] T. Tang, Moving mesh methods for computational fluid dynamics, Contemporary Math. 383 (2005) 141–173.

[40] H. D. Ceniceros, T. Y. Hou, An efcient dynamically adaptive mesh for potentially singular solutions, Journal of Computational Physics 172 (2001) 609–639.

[41] E. Walsh, Moving mesh methods for problems in meteorology, Ph.D. thesis, University of Bath (2010).

[42] C. Budd, M. Cullen, E. Walsh, Monge-ampère based moving mesh methods for numerical weather prediction, with applications to the Eady problem, J. of Comp. Phys. 236 (2013) 247–270.

[43] W. Huang, R. D. Russell, A moving collocation method for solving time dependent partial differential equations, Appl. Num. Math. 20 (1996) 101–116.

[44] G. Beckett, J. Mackenzie, Convergence analysis of finite difference approximations on equidistributed grids to a singularly perturbed boundary value problem, Appl. Num. Math. 35 (2000) 87–109.

[45] W. Huang, R. D. Russell, Analysis of moving mesh partial differential equations with spatial smoothing, J. Comp. Phys. 34 (1997) 1106–1126.

[46] B. Edmonstone, O. Matar, R. Craster, Coating of an inclined plane in the presence of insoluble surfactant, J. Colloid Interface Sci. 287 (2005) 261–272.

[47] B. Edmonstone, O. Matar, R. Craster, A note on the coating of an inclined plane in the presence of soluble surfactant, J. Colloid Interface Sci. 293 (2006) 222–229.

[48] L. Zhornitskaya, A. L. Bertozzi, Positivity preserving numerical schemes for lubrication-type equations, SIAM J. Num. Anal. 37 (2000) 523 – 555.

[49] W. Huang, Y. Ren, R. D. Russell, Moving mesh methods based on moving mesh partial differential equations, Comp. Phys. 113 (1994) 279–290.

[50] L. R. Petzold, A description of DASSL: A differential/algebraic system solver, Tech. rep., Sandia Labs (1982).

[51] P. Brown, A. Hindmarsh, L. Petzold, Using Krylov methods in the solution of large-scale differential- algebraic systems, SIAM J. Sci. Comp. 15 (1994) 1467–1488.

[52] A. Bertozzi, M. Brenner, Linear stability and transient growth in driven contact lines, Phys. Fluids 9 (3) (1997) 530–539.